

2분반 8팀 : 책사조
박지우 김도연 나하윤 성지운



북 치고 중고 치고

목차

1

개요 및 특징

2

세부 기능

3

시스템 구조 및 구성 요소

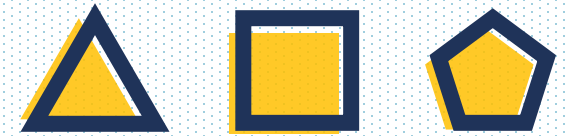
4

구현 방법

5

팀원 업무 분담 및 소감

Part 1



개요 및 특징

Part 1

개요 및 특징

개발이유 및 목적

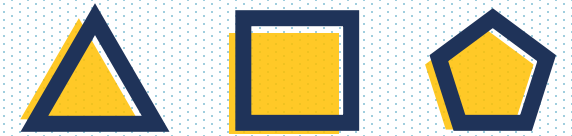
전공교재의 가격이 부담되는 대학생들을 위해

중고책 거래와 그 교재와 연계된 스터디 모집 서비스를 제공하는 시스템이 필요하다고 생각함.

더불어 기존의 서비스와 차별점을 둘 수 있는 특별한 기능들을 더 추가하여 개발하고자 함.



Part 2



세부기능

대분류	기능
로그인/회원가입	로그인 기능, 회원가입 기능
	MY 정보 확인 기능, 로그아웃 기능
	회원 탈퇴 기능
스터디 그룹	스터디 모집 확인 기능
	스터디 모집 등록 기능
	스터디 참가 기능 (그룹 채팅 이용)
회원간 채팅	MY 채팅 목록 확인 기능
	채팅 방 입장 기능
	그룹 채팅(스터디 그룹끼리) 기능
	개인 채팅(판매자 및 구매자) 기능
중고책 판매	판매 글 등록, 수정, 삭제, 거래 상태 수정, 검색 기능
	즐거찾기 등록 및 조회 기능
	주문 완료 품목 및 판매 품목 회원 별 조회 기능
중고책 경매	경매 글 작성, 조회, 수정, 삭제, 검색 기능
	자신의 경매 글 및 입찰 리스트 조회 (결과, 진행상황 조회 한번에 가능)
	경매 입찰, 낙찰 기능: 낙찰 시 낙찰자/판매자 정보는 알림 창에서 확인 가능
도서 공동구매	공구 게시글 등록(모집), 수정, 삭제, 검색, 상세조회
	전체 공구 게시글 목록보기, 내가 등록한 공구 게시글 목록보기, 내가 신청한 공구 게시글 목록보기
	공구 참여신청, 신청취소

Part 2

세부 기능

회원가입 / 로그인

북치고 중고치고

아이디

비밀번호

Login

Join

회원가입

아래 정보들을 입력해주세요!

아이디

이메일

비밀번호

비밀번호 확인

이름

주소

전화번호를 입력해주세요.

계좌 번호를 입력해주세요.

회원가입

북치고 중고치고
MY 정보

회원 탈퇴

정보 내용

아이디 dongduk2021

이름 김동덕

계좌번호 <279-111111-11111

전화번호 010-1111-1111

주소 서울시 성북구

거래횟수 0

스터디 등록

스터디 이름

토론 일정

도서 이름

작가

스터디 소개

사진

정원

파일 선택

선택된 파일 없음

5명

등록하기

스터디원 모집

스터디 등록



스터디1번 클릭 시 채팅방 이동

1/5

토론일정: 내일 7시

도서정보: 박지우의 도서 (글쓴이 : 박지우)

안녕하세요 박지우의 도서입니다. 많이 참가해주세요!!

채팅방

채팅방 화면 예시

nn: 안녕하세요 하운님!
2021-05-19 19:42:35.388

안녕하세요
2021-05-19 22:32:09.424

nn: 혹시 제 알 보이시나요??
2021-05-19 22:32:11.147

nn: 넵넵 안녕하세요!
2021-05-19 22:32:13.215

nn: 혹시 제 알 잘 보이시나요 하운님?
2021-05-19 22:32:21.312

네네 잘보여요!!
2021-05-19 22:32:26.268

메시지를 입력하세요.

< 회원 간 채팅 화면 >

회원 관련 기능

로그인, 회원가입, MY 정보 확인 기능

스터디 그룹 기능

스터디 그룹 등록 작성 기능, 목록 조회
스터디 방 입장 시 그룹채팅 기능

Part 2

세부 기능

중고책 판매 기능

즐거찾기 조회

주문 완료 조회

판매글 조회

판매 등록

검색

창문 넘어 도망친 100세 노인

15000원

해리포터와 마법사의 돌

10000원

판매글 목록 조회

판매글 전체 리스트 정렬
즐거찾기, 주문 완료, 판매 글 조회 가능
도서 이름으로 검색 가능

주문 완료 조회				
번호	사진	제목	가격	
1		해리포터와 마법사의 돌	10000	

즐거찾기 조회				
번호	사진	제목	가격	-
1		창문 넘어 도망친 100세 노인	15000	<div>삭제</div>

판매글 조회				
번호	사진	제목	가격	거래 상태
1		창문 넘어 도망친 100세 노인	15000	판매 중

각종 조회 페이지

주문 완료 조회, 즐거찾기 조회,
판매글 조회

판매 등록

상품 이름

가격

저자

출판사

상품 정보

사진 등록

파일 선택

선택된 파일 없음

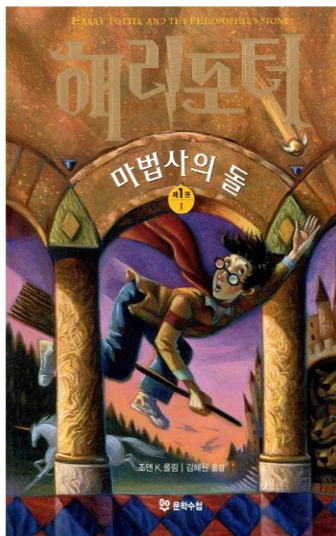
거래 방식

등록

취소

판매 게시글 작성

판매 글 작성 기능.
수정도 같은 페이지



해리포터와 마법사의 돌

10000원

저자 J.K. 롤링

출판사 문학수집

거래 방식 우체국 택배

거래 상태 판매 중

거래 상태 수정

계시글 삭제

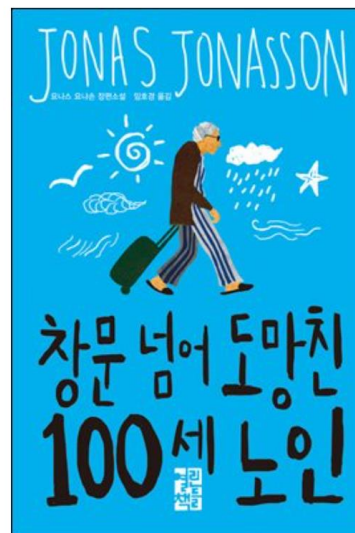
계시글 수정

상품 정보

보관 잘 해서 상태 좋습니다.

판매글 상세 페이지 (판매자)

거래 상태 수정 기능 : 거래 상태 수정 페이지에서
구매자 아이디로 상품 주문 완료 처리 가능
판매글 수정 기능 : 판매글 등록폼과 형식 동일
판매글 삭제 기능



창문 넘어 도망친 100세 노인

15000원

저자 요나스 요나손

출판사 열린책들

거래 방식 편의점 택배

거래 상태 판매 중



판매자 :

김자바

판매자 거래 횟수 :

1

채팅하기

상품 정보

약간의 변색 말고는 상태 좋아요!

판매글 상세 페이지 (구매자)

하트 버튼 클릭 시 즐겨찾기 등록
채팅 페이지로 이동 가능

Part 2

세부 기능

중고책 경매 기능

도서 이름이, 작가 이름만 검색 가능합니다.

☒ 도서 이름 ☐ 작가 이름

경매 게시글 작성

올린 경매 조회

입찰 내역 조회

최근 경매 목록



D - 4
머리부터 천천히
박슬외 | 등적출판
20 원



D - 2
스프링 부트로 배우는
자바 웹 개발
박복한사범 | 등적출판
22 원



D - 31
다시 또 여름
이여름 | 등적출판
1002 원



D - 101
소년이 온다
한강 | 등적출판
300 원



D - 70
나의 라임 오렌지 나무
바스콘셀루스 | 상상출판
1 원



경매 목록 조회

경매 전체 리스트 (최신순) 정렬
경매글 작성, 올린 경매 및 입찰 내역 조회 가능
경매글 도서, 작가 이름으로 검색 가능

"소년이 온다" 검색 결과입니다.



D - 101
소년이 온다
한강 | 등적출판
300 원

경매 목록 가기

경매 도서 검색 기능

도서 및 작가이름으로 경매글 검색

▼ 이미지 첨부 ▼

파일 선택

책 제목

출판사

지은이

연도-월-일

상태

시작 가격

경매 올리기

취소

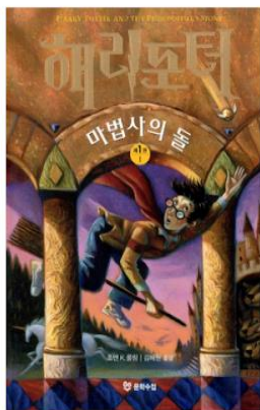
경매 게시글 작성

경매 글 작성 기능.
수정도 같은 페이지

Part 2

세부 기능

중고책 경매 기능



해리포터 초판

조앤K롤링 | 상상출판

마감 날짜 : 2021년 06월 26일

도서 상태 : 매우 최상

가격 시작 : 2000원

경매 마감

게시글 수정

경매 목록

경매 삭제

1103 님
10000원

1102 님
9000원

1101 님
5000원

1100 님
2001원

경매글 조회

경매자 : 게시글 수정/삭제/낙찰(마감) 가능
입찰자 : 경매 입찰 가능 및 입찰금액 순위 확인

15000

입찰하기

취소

입찰이 완료되었습니다.

입찰자 정보

입찰 id

1105

입찰자 성함

김자바

경매 고유 id

2108

경매 아이템

머리부터 천천히

입찰 가격

800

입찰 하기

시작금액 이상에서 입찰이 가능

"머리부터 천천히" 경매가 마감되었습니다.

입찰자 정보

낙찰 id

1108

낙찰자 성함

김자바

경매 아이템

머리부터 천천히

낙찰 가격

15000

경매 <머리부터 천천히> 낙찰자 정보입니다.

<경매자 알림창>

경매 <머리부터 천천히> (고유 id: 2108) 마감하였습니다.
최종 낙찰 가격은 <15000원> 입니다.
낙찰자는 <김자바> 님이며 핸드폰 번호는 <010-2222-2222>, 주소는 <서울시 성북구>로 입니다.
낙찰자에게도 해당 고객님의 정보가 전송되었습니다. 감사합니다.

경매 <해리포터> 낙찰 되었습니다.

경매 <해리포터> (고유 id: 2061)에 낙찰 되었습니다.
낙찰자 <지운> 님은 해당 계좌번호 <222-111111-11111>로 입금 금액 <2001원>을 넣어주세요.
판매자에게도 해당 고객님의 정보가 전송되었습니다. 축하드립니다.

<낙찰자 알림창>

경매자 : 낙찰하기

마감 버튼 클릭시 낙찰 바로 진행

가장 높은 금액이 낙찰, 낙찰자 정보는 알림창에서 확인 가능
낙찰자 또한 알림창에서 경매자 계좌번호 확인 가능

Part 2

세부 기능

도서 공동 구매 기능

전체 목록

전체 목록보기

모집하기

나의 모집

나의 신청

고유번호	책	정보	공구가	공구원
57		<리얼 제주> 공구합니다 ISBN : 38888 모집자 : java2021 기한: 2021-08-19까지	20000	0 / 21
55		<도큐클:re> 이번달까지 공구합니다 ISBN : 1999 모집자 : hy 기한: 2021-06-30까지	51000	0 / 30
56		<하루 30분, 그림책 놀이> 힐링책 공구해요 ISBN : 3789 모집자 : nn 기한: 2021-07-07까지	15000	0 / 15

검색어를 입력하세요

search

<나의 구매 신청목록>

나의 모집 목록

전체 목록보기

모집하기

나의 모집

나의 신청

고유번호	책	정보	공구가	공구원
57		<리얼 제주> 공구합니다 ISBN : 38888 모집자 : java2021 기한: 2021-08-19까지	20000	0 / 21
58		<오만과 편견> 공구합니다 ISBN : 1234 모집자 : java2021 기한: 2021-06-26까지	17000	0 / 17

<'그림책' 검색 결과>

고유번호	책	정보	공구가	공구원
56		<하루 30분, 그림책 놀이> 힐링책 공구해요 ISBN : 3789 모집자 : nn 기한: 2021-07-07까지	15000	0 / 15

검색어를 입력하세요

search

공동구매 모집하기

모집 시작하기

파일 선택 오만과...jpg

ISBN : 1234

제목 : 오만과 편견

저자 : 제인 오스틴

출판사 : 위즈덤하우스

가격 : 22000

책 줄거리 : 제인 오스틴의 사랑과 결혼

게시글 고유번호

고유번호 미정, 게시글 등록완료시 고유번호가 부여됩니다.

제목

<오만과 편견> 공구합니다

모집자

java2021

모집기한

2021-06-26

공동구매 목록 조회

공동구매 전체 리스트 정렬
나의 공동구매글, 구매 신청 목록 조회 가능
구매글 도서 이름으로 검색 가능

각종 조회 기능

나의 공동구매글 목록 조회 및
도서 이름으로 공동구매 글 검색

구매 게시글 작성

공동구매 글 작성 기능.
수정도 같은 페이지

게시글 보기

참여할 수 있는 공구입니다.

공구 참여하기



ISBN : 1999

제목 : 도쿄구울:re

저자 : 이시다 스이

출판사 : 대원씨아이

가격 : 60000

책 줄거리 : 카네키 켄의 CCG 이야기

신청한 사람

이 공동구매의 리더만 볼 수 있습니다.

게시글 고유번호

55

게시글 제목

<도쿄구울:re> 이번달까지 공구합니다

공동 구매글 조회

작성자 : 게시글 수정/삭제/마감 가능
구매 참여자 : 공구 신청 가능

게시글 보기

이미 신청한 공동구매입니다.

신청 취소하기



ISBN : 1999

제목 : 도쿄구울:re

저자 : 이시다 스이

출판사 : 대원씨아이

가격 : 60000

책 줄거리 : 카네키 켄의 CCG 이야기

신청한 사람

이 공동구매의 리더만 볼 수 있습니다.

게시글 고유번호

55

구매 신청 시 뜨는 화면

'이미 신청한 공동구매입니다' 알림 뜸

나의 신청 목록

전체 목록보기 모집하기 나의 모집 나의 신청

고유번호	책	정보	공구가	공구원
55		<도쿄구울:re> 이번달까지 공구합니다 ISBN : 1999 모집자 : hy 기한: 2021-06-30까지	51000	1 / 30

검색어를 입력하세요

search

나의 구매 신청 목록

공동구매 메인 페이지에서 나의 구매
신청 목록 확인 가능

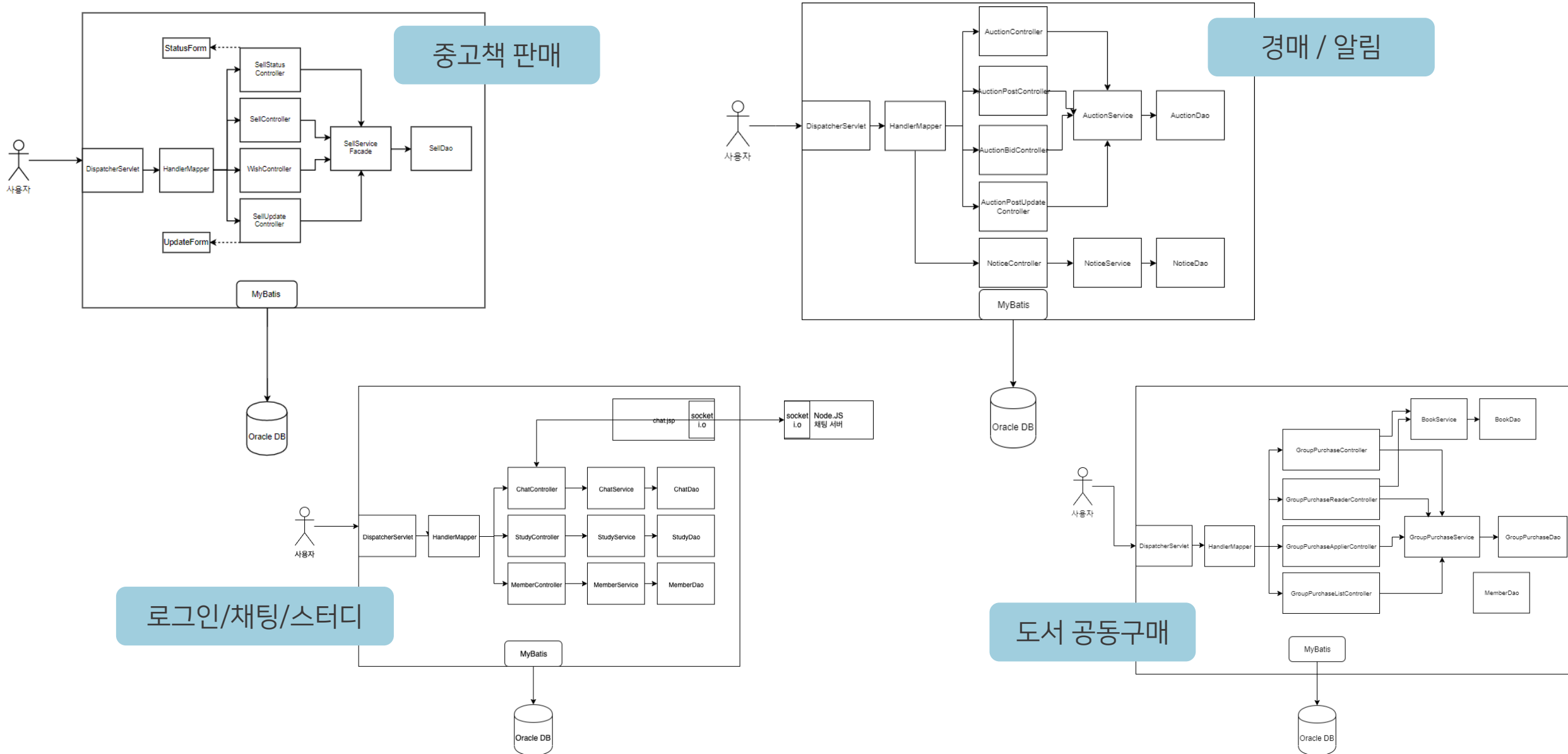
Part 3



시스템 구조 및 구성 요소

Part 3

시스템 구조 : 시스템 구성도



Part 3

세부구성요소 (Domain)

이름	설명
SellItem.java	판매 아이템 관련 클래스
Wishlist.java	위시리스트 관련 클래스
BuyFin.java	주문 완료 아이템 관련 클래스
Chat.java	채팅 관련 클래스
ChatMessage.java	채팅 메시지 관련 클래스
ChatRoom.java	채팅 방 관련 클래스
Member.java	회원 관련 클래스

이름	설명
Study.java	스터디 관련 클래스
GroupPurchase.java	공동구매 게시글 관련 클래스
GpApply.java	누가, 어떤 공동구매를 신청 했는 지와 관련된 클래스
Book.java	책 정보 관련 클래스
AuctionBid.java	경매 입찰 관련 클래스
AuctionItem.java	경매 글 관련 클래스
Notice.java	알림 창 관련 클래스

Part 3

세부구성요소 (Controller)

멤버 관련 Controller

이름	설명
MemberController.java	Member와 관련된 전반적인 기능 수행

채팅 관련 Controller

이름	설명
ChatController.java	채팅과 관련된 전반적인 기능 수행

스터디 관련 Controller

이름	설명
StudyController.java	스터디 방 관련된 전반적인 기능 수행

중고 판매 관련 Controller

이름	설명
SellController.java	중고 판매와 관련된 기본적인 기능 수행
SellStatusController.java	판매 품목 거래 상태 변경과 관련된 기능 수행
SellUpdateController.java	판매 품목 정보 변경과 관련된 기능 수행
WishController.java	위시리스트와 관련된 기능 수행
StatusForm.java	SellStatusController에 사용되는 Command 객체
UpdateForm.java	SellUpdateController에 사용되는 Command 객체

알림 관련 Controller

이름	설명
NoticeController.java	알림 관련 기능 수행

공동구매 관련 Controller

이름	설명
GroupPurchaseListController.java	공동구매와 관련된 리스트를 불러오는 기능 수행
GroupPurchaseController.java	공동구매 게시글 상세보기 기능 담당
GroupPurchaseApplicerController.java	공동구매 신청과 취소 기능 수행
GroupPurchaseReaderController.java	공동구매 게시글 등록, 수정, 삭제 기능 수행

경매 관련 Controller

이름	설명
AuctionBidController.java	경매 입찰하기, 낙찰하기 기능 수행
AuctionController.java	경매 글 및 입찰 목록 불러오기, 경매 글 검색 수행
AuctionPostController.java	경매 게시글 작성관련 기능 수행
AuctionPostUpdateController.java	경매 게시글 수정 관련 기능 수행

Part 3

세부구성요소 (DAO + Service)

멤버

이름
MemberDao.java
MemberMapper.java
MemberMapper.xml
MemberServiceImpl.java

스터디

이름
StudyDao.java
StudyMapper.java
StudyMapper.xml
StudyServiceImpl.java

알림

이름
NoticeDao.java
MyBatisNoticeDao.java
NoticeMapper.java
NoticeMapper.xml
NoticeService.java
NoticeServiceImpl.java

공동 구매

이름
GroupPurchaseDao.java
MyBatisGroupPurchaseDao.java
GroupPurchaseMapper.java
GroupPurchaseMapper.xml
GroupPurchaseService.java
GroupPurchaseServiceImpl.java

책 : 게시글에 등록된 책의 목록들

이름
BookDao.java
MyBatisBookDao.java
BookMapper.java
BookMapper.xml
BookService.java
BookServiceImpl.java

경매

이름
AuctionDao.java
MyBatisAuctionDao.java
AuctionMapper.java
AuctionMapper.xml
AuctionService.java
AuctionServiceImpl.java

중고 판매

이름
SellDao.java
MyBatisSellDao.java
SellMapper.java
SellMapper.xml
SellServiceFacade.java
SellServiceImpl.java

채팅

이름
ChatDao.java
ChatMapper.java
ChatMapper.xml
ChatServiceImpl.java

Part 3

세부구성요소 (JSP)

각 폴더 별 jsp 파일들

member

이름	설명
login.jsp	로그인 페이지
my.jsp	내 정보 페이지
register.jsp	회원가입 페이지

chat

이름	설명
list.jsp	채팅 목록 페이지
chat.jsp	채팅 방 페이지

study

이름	설명
list.jsp	스터디 목록 페이지
enroll.jsp	스터디 등록 페이지

sell

이름	설명
sell_finish.jsp	주문 완료 조회 페이지
sell_history.jsp	판매 글 조회 페이지
sell_list.jsp	전체 판매 글 리스트 조회 페이지
sell_publisher.jsp	구매자 시점의 판매 품목 상세 페이지
sell_seller.jsp	판매자 시점의 판매 품목 상세 페이지
sell_status.jsp	(판매자 시점) 판매 품목 거래 상태 변경 페이지
sell_update.jsp	(판매자 시점) 판매 품목 정보 변경 페이지
sell_upload.jsp	판매 글 등록 페이지
sell_wishlist.jsp	전체 위시리스트 보기 페이지

notice

이름	설명
notice_list.jsp	알림 창 목록 페이지
notice_post.jsp	알림 상세히 보기 페이지

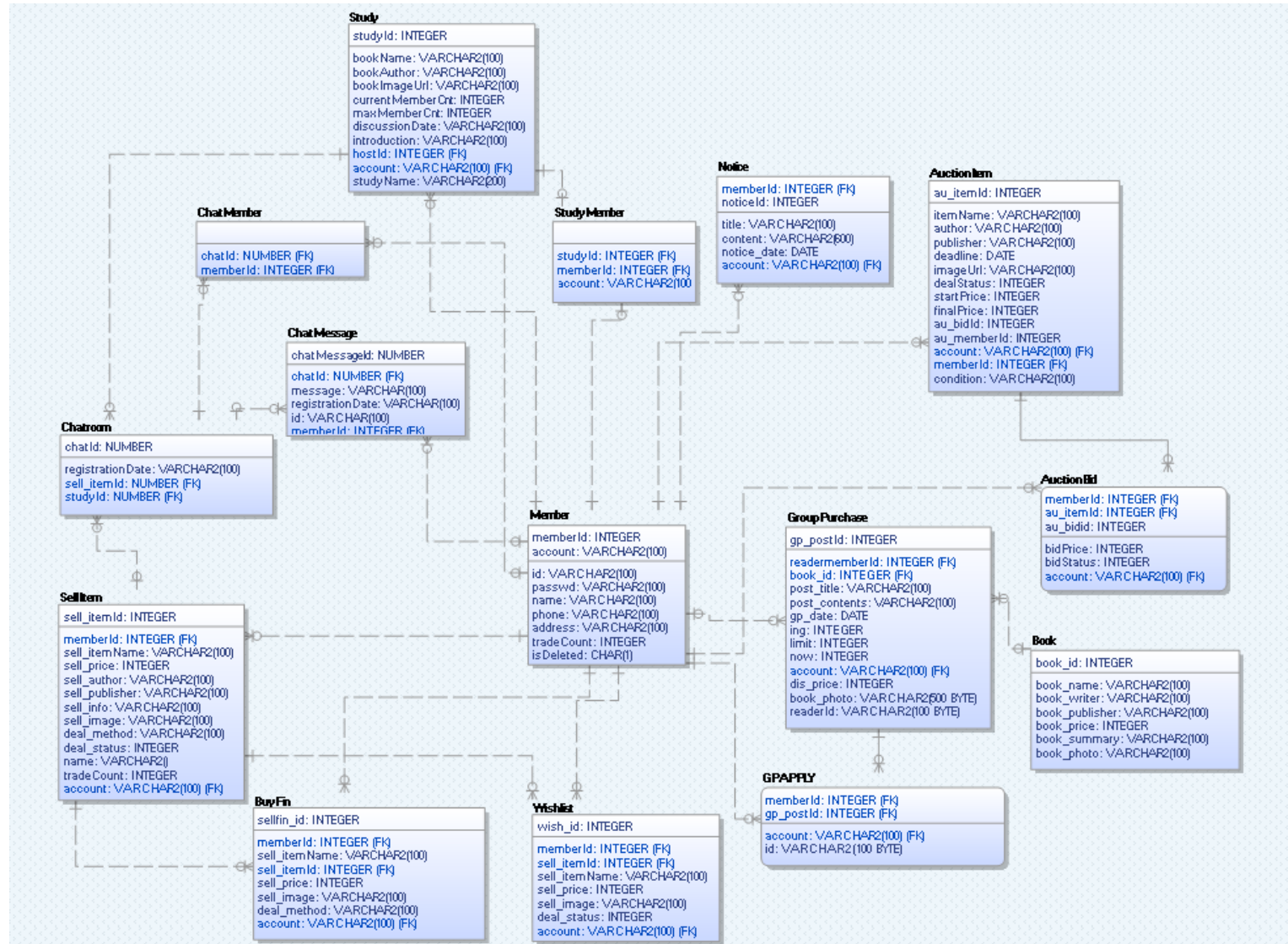
groupPurchase

이름	설명
gp_detail_applier.jsp	게시글 상세 페이지 (자신이 이미 신청한 공구일 때)
gp_detail_finish_reader.jsp	게시글 상세 페이지 (자신이 등록한 게시글 && 이미 마감된 공구)
gp_detail_finish.jsp	게시글 상세 페이지 (이미 마감된 공구일 때)
gp_detail_reader.jsp	게시글 상세 페이지 (자신이 등록한 게시글일 때)
gp_detail.jsp	게시글 상세 페이지 (신청가능한 공구일 때)
gp_list_applier.jsp	자신이 신청한 공구 게시글 리스트 보기
gp_list_reader.jsp	자신이 등록한 공구 게시글 리스트 보기
gp_list.jsp	전체 공구 게시글 보기
gp_update.jsp	공구 게시글 수정 페이지
gp_upload.jsp	공구 게시글 등록을 위한 입력(form) 페이지

auction

이름	설명
auction_bidApply.jsp	경매 입찰 작성 페이지
auction_bidComplete.jsp	경매 입찰 완료 시 확인 페이지
auction_delete_success.jsp	경매 글 삭제 완료 시 확인 페이지
auction_itemClose.jsp	경매 마감(낙찰) 시 뜨는 낙찰자 정보 알려주는 페이지
auction_list_auctioneer.jsp	자신의 경매 글 목록 보기
auction_list_bidder.jsp	자신의 입찰 목록 보기
auction_list_search.jsp	경매 글 검색 시 결과 페이지
auction_read.jsp	경매 글 상세히 조회 페이지
auction_update.jsp	경매 글 수정 페이지
auction_upload.jsp	경매 글 작성 페이지

Database Schema



Part 3

기능별 Request 처리 흐름

중고 판매 request
처리 흐름

기능	세부기능	입력화면	request URI	request parameters	Controller class & request handler method	Service Class & method	DAO class & method	model data(이름, 객체 타입)	출력화면	출력방법								
판매 리스트 페이지로 이동	main/main.jsp		/sell/list.do	-	SellController.getList()	SellServiceImpl.getList()	SellDao.getList()	("list", List<SellItem>)	/sell/sell_list.jsp	forwarding								
	판매 리스트에서 판매글 클릭 시 세부 정보 불러오기	/sell/sell_list.jsp (by 판매자 시점)	/sell/item.do	sell_itemId (memberId는 session으로부터 가져옴)	SellController.getItem()	SellServiceImpl.getItemById() - memberId 알고 국가명 세션의 memberId랑 비교, 동일하면 판매자 페이지 호출 SellServiceImpl.getItemByItemId() - sellItem 객체 읽어옴, SellServiceImpl.getCountById() - id를 tradeCount 값 읽어온 후 sellItem 객체에 set	SellDao.getItemById(), SellDao.getItemByItemId(), SellDao.getCountById()	("item", SellItem), ("alert", "X")	/sell/sell_seller.jsp	forwarding								
	/sell/sell_list.jsp (by 구매자 시점)	/sell/item.do		sell_itemId (memberId는 session으로부터 가져옴)	SellController.getItem()	SellServiceImpl.getItemById() - memberId 알고 국가명 세션의 memberId랑 비교, 동일하면 구매자 페이지 호출 SellServiceImpl.getItemByItemId() - sellItem 객체 읽어옴, SellServiceImpl.getCountById() - id를 tradeCount 값 읽어온 후 sellItem 객체에 set	SellDao.getItemById(), SellDao.getItemByItemId(), SellDao.getCountById()	("item", SellItem), ("alert", "X")	/sell/sell_publisher.jsp	forwarding								
직 이름으로 판매 게시물 검색	/sell/sell_list.jsp	/sell/search.do		sell_itemName	SellController.searchList()	SellServiceImpl.getSellByItemName()	SellDao.getSellByItemName()	("list", List<SellItem>)	/sell/sell_list.jsp	forwarding								
판매글 조회 페이지로 이동	/sell/sell_list.jsp	/sell/history.do		- (memberId session으로부터 가져옴)	SellController.getHistory()	SellServiceImpl.getSellByMembr		거래 상태 수정 페이지로 이동	/sell/sell_seller.jsp	/sell/statusUpdate.do(GET)	sell_itemId	SellStatusController.showForm()	-	-	/sell/sell_status.jsp	forwarding		
	주문 완료 조회 페이지로 이동	/sell/sell_list.jsp	/sell/finish.do		- (memberId session으로부터 가져옴)	SellController.buyFin()	SellServiceImpl.selectBuyFin()		거래 상태 수정	/sell/sell_status.jsp (수정 버튼 클릭 시)	/sell/statusUpdate.do(POST)	- (커먼드 객체 사용)	SellStatusController.update()	SellServiceImpl.updateDealStatus(), SellServiceImpl.getMemberById(), SellServiceImpl.getSellByItemId(), SellServiceImpl.selectAllWishlist(), SellServiceImpl.insertBuyFin()	SellDao.updateDealStatus(), SellDao.getMemberById(), SellDao.getSellByItemId(), SellDao.insertBuyFin(), SellDao.selectAllWishStatus(), SellDao.updateWishStatus()	-	/sell/item.do?sell_itemId=	redirecting
판매 등록 페이지로 이동	/sell/sell_list.jsp	/sell/uploadForm.do		-	mvc-view-controller (정적 view 제공)	-		게시글 수정	/sell/sell_status.jsp(취소 버튼 클릭 시)	/sell/item.do	sell_itemId (memberId는 session으로부터 가져옴)	SellController.getItem()	SellServiceImpl.getItemById(), SellServiceImpl.getSellByItemId(), SellServiceImpl.getCountById()	SellDao.getItemById(), SellDao.getSellByItemId(), SellDao.getCountById()	("item", SellItem), ("alert", "X")	/sell/sell_seller.jsp	forwarding	
판매글 등록	/sell/sell_upload.jsp (등록 시)	/sell/upload.do	sell_itemName, sell_price, sell_author, sell_publisher, sell_info, deal_method (memberId는 session으로부터 가져옴)	SellController.upload()	SellServiceImpl.insertSellItem(), SellServiceImpl.getSellList(), SellServiceImpl.updateCountPlus()			게시글 수정 페이지로 이동	/sell/sell_seller.jsp (거래 상태 판매 중이 아닌 경우)	/sell/update.do(GET)	sell_itemId	SellUpdateController.showForm()	SellServiceImpl.getDealStatus(), SellServiceImpl.getSellByItemId(), SellServiceImpl.getCountById()	SellDao.getDealStatus(), SellDao.getSellByItemId(), SellDao.getCountById()	("item", SellItem)	/sell/sell_update.jsp	forwarding	
	/sell/sell_upload.jsp (취소 시)	/sell/list.do	-		SellController.getList()	SellServiceImpl.getSellList()			게시글 수정	/sell/sell_update.jsp (수정 버튼 클릭 시)	/sell/update.do(POST)	- (커먼드 객체 사용)	SellUpdateController.update()	SellServiceImpl.updateSellItem(), SellServiceImpl.getSellByItemId(), SellServiceImpl.selectAllWishlist(), SellServiceImpl.updateWishStatus(), SellServiceImpl.deleteSellItem(), SellServiceImpl.getSellList(), SellServiceImpl.getMemberById(), SellServiceImpl.updateCountMinus()	SellDao.updateSellItem(), SellDao.getSellByItemId(), SellDao.selectAllWishlist(), SellDao.updateWishStatus(), SellDao.deleteSellItem(), SellDao.getSellList(), SellDao.getMemberById(), SellDao.updateCountMinus()	-	/sell/item.do?sell_itemId=	redirecting
	/sell/sell_upload.jsp (취소 시)			-	SellController.getList()	SellServiceImpl.getSellList()			게시글 삭제	/sell/sell_seller.jsp (확인 버튼 클릭 시)	/sell/delete.do	sell_itemId	SellController.delete()	SellServiceImpl.deleteSellItem(), SellServiceImpl.getSellList(), SellServiceImpl.getItemById(), SellServiceImpl.updateCountMinus()	SellDao.deleteSellItem(), SellDao.getSellList(), SellDao.getItemById(), SellDao.updateCountMinus()	("list", List<SellItem>)	/sell/sell_list.jsp	forwarding
									즐거찾기 조회 페이지로 이동	/sell/sell_list.jsp	/sell/wish.do	- (memberId session으로부터 가져옴)	WishController.getList()	SellServiceImpl.selectAllWishlist()	SellDao.selectAllWishlist()	("list", List<Wishlist>)	/sell/sell_wishlist.jsp	forwarding
									즐거찾기 등록	/sell/sell_publisher.jsp	/sell/wish/upload.do	sell_itemId, sell_itemName, sell_price, sell_image, deal_status (memberId는 session으로부터 가져옴)	WishController.upload()	SellServiceImpl.insertWishlist(), SellServiceImpl.getSellByItemId(), SellServiceImpl.getCountById()	SellDao.insertWishlist(), SellDao.getSellByItemId(), SellDao.getCountById()	("item", SellItem)	/sell/sell_publisher.jsp	forwarding
									즐거찾기 삭제	/sell/sell_wishlist.jsp	/sell/wish/delete.do	wish_id	WishController.delete()	SellServiceImpl.deleteWishlist(), SellServiceImpl.selectWishlist()	SellDao.deleteWishlist(), SellServiceImpl.selectWishlist()	("list", List<Wishlist>)	/sell/sell_wishlist.jsp	forwarding
채팅하기 페이지로 이동	/sell/sell_publisher.jsp (채팅 가능 상태일 경우)	/sell/chat.do		sell_itemId	SellController.chat()			채팅하기 페이지로 이동	/sell/sell_publisher.jsp (채팅 불가 상태일 경우)	/sell/chat.do	sell_itemId	SellController.chat()	SellServiceImpl.getDealStatus(), SellServiceImpl.getSellByItemId(), SellServiceImpl.getCountById()	SellDao.getDealStatus(), SellDao.getSellByItemId(), SellDao.getCountById()	("item", SellItem), ("alert", "채팅은 상품이 판매 중 상태일 때만 가능합니다.")	/sell/sell_publisher.jsp	forwarding	

중고 판매 request 처리 흐름

Part 3

기능별 Request 처리 흐름

로그인, 회원가입,
MY 정보 보기
request 처리 흐름

로그인 및 회원가입	로그인 페이지	member/login.jsp	member/login	-	MemberController.login()	MY 정보 보기 request 처리 흐름				-	member/login.jsp	forwarding
	로그인 시도	member/login.jsp	member/login.do	String username, String password	MemberController.login(String id, String password)	MemberServiceImpl.login(String id, String password)	memberDao.getPassword(username) memberDao.getAccount(username) memberDao.getMemberid(username)	["id" : 아이디, "password" : 암호]	main.jsp	forwarding		
	회원 가입 페이지	member/register.jsp	member/register	-	MemberController.register()	-	-	-	member/register.jsp	forwarding		
	회원 가입 시도	member/register.jsp	member/register.do	Member member	MemberServiceImpl.registerDo(Member member) {	memberService.insertMember(member)	memberDao.insertMember(member)	["member" : member]	member/login.jsp	forwarding		
	로그인	member/login.jsp	member/login.do	-	MemberController.login(String id, String password)	MemberServiceImpl.login(String id, String password)	MemberDao.isValidUser(String id, String password)	["id" : 아이디, "password" : 암호]	main.jsp	forwarding		
	로그아웃	main.jsp	member/logout.do	-	String logoutDo(HttpServletRequest request)	-	-	-	main.jsp	redirect		
	회원탈퇴	main.jsp	member/delete.do	-	String request	스터디 목록	study/list.jsp	study	-	ModelAndView list()	studyService.getStudyList()	
	MY정보 보기	member/my.jsp	member/my	-	Mode request	스터디 모집 등록 페이지	study/list.jsp	study/register	-	String register()	-	

스터디, 채팅
request 처리 흐름

스터디	스터디 목록	study/list.jsp	study	-	ModelAndView list()	studyService.getStudyList()	studyDao.getStudyList()	{list: List<Study>}	study/list.jsp	forwarding
	스터디 모집 등록 페이지	study/list.jsp	study/register	-	String register()	-	-	-	study/enroll.jsp	forwarding
	스터디 모집 등록 요청	study/enroll.jsp	study/register.do	Study study, MultipartFile file	String registerDo(HttpServletRequest request, Study study, MultipartFile file)	studyService.insertStudy(study)	studyDao.insertStudy(study)	-	study/list.jsp	forwarding
채팅	채팅 목록	main.jsp	chat	-	ModelAndView list(HttpServletRequest request)	chatService.getChatRoomList(memberid)	chatDao.getChatRoomList(memberid)	{"list": List<Chat>}	chat/list.jsp (클릭 시)	forwarding
	채팅방 페이지	chat/list.jsp	chat/room	String sellid, String studyid	ModelAndView room(HttpServletRequest request, @RequestParam(required = false, value="sellid") String sellid, @RequestParam(required = false, value="studyid") String studyid)	chatService.getChatBySellid(sellidint) chatService.insertChatMember(chat.getChatid(), memberid) chatService.getChatByStudyid(studyidint) chatService.insertChatMember(chatid, memberid) chatDao.getSellerid(chat.getSell_itemid())	chatDao.getChatBySellid(sell_itemid) chatDao.getChatByStudyid(studyid) chatDao.insertChatRoom(chat) chatDao.insertChatMember(chatid, memberid) chatDao.getSellerid(chat.getSell_itemid())	{"chatRoomId": id, "chatid": chatid}	chat/chat.jsp	forwarding
	채팅 메시지 입력	chat/chat.jsp	chat/insert.do	ChatMessage message	String insert(@RequestBody ChatMessage message)	chatService.insertChatMessage(message)	chatDao.insertChatMember(chatid, memberid)	-	-	-

Part 3

기능별 Request 처리 흐름

공동구매 request 처리 흐름

공동구매

공동구매

공통구매 request
처리 흐름

전체 공동구매 리스트 페이지로 이동	/main/main.jsp	/groupPurchase/list.do	-	GroupPurchaseListContoller.getPostList() GroupPurchaseServiceImpl.update(int gp_postid)	GroupPurchaseDao.getGroupPurchaseList() GroupPurchaseDao.update(int gp_postid)	("PostList", List<GroupPurchase>)	/groupPurchase/gp_list.jsp	forwarding						
내가 모집하는 공유리스트 보기	/groupPurchase/gp_list.jsp	/groupPurchase/mReader/list.do	-	GroupPurchaseListContoller.getMyPostList(HttpServletRequest request)	GroupPurchaseServiceImpl.getGroupPurchaseMemberList(int readerMemberId)	GroupPurchaseServiceDao.getGroupPurchaseMemberList(int readerMemberId)	/groupPurchase/gp_list_reader.jsp	forwarding						
내가 신청한 공유리스트 보기	/groupPurchase/gp_list.jsp	/groupPurchase/mApplier/list	-	GroupPurchaseListContoller.getMyApplyList(HttpServletRequest request)	GroupPurchaseServiceImpl.getGpPostByPostId(int gp_postid)	GroupPurchaseDao.getGpPostByPostId(int gp_postid)	/groupPurchase/gp_list_applier.jsp	forwarding						
해당책명과 관련된 공유리스트 보기	/groupPurchase/gp_list.jsp	/groupPurchase/search.do	String book_name	GroupPurchaseListContoller.getSearchList(@RequestParam("book_name") String book_name)	GroupPurchaseServiceImpl.Search_GroupPurchase(String book_name)	GroupPurchaseDao.Search_GroupPurchase(String book_name)	/groupPurchase/gp_list.jsp	forwarding						
리스트에서 클릭시 게시글 자세히 보기 가능	/groupPurchase/gp_list.jsp /groupPurchase/gp_list_reader.jsp /groupPurchase/gp_list_applier.jsp	/groupPurchase/detail.do	int gp_postid	GroupPurchaseController.detailPostToServletRequest request, @RequestParam("gp_postid") int gp_postid)	(본인작성) 공동구매 모집글 수정 페이지로 이동	/groupPurchase/gp_detail_reader.jsp	/groupPurchase/updateForm.do	int gp_postid	GroupPurchaseReaderController.updateForm(HttpServletRequest request, @RequestParam("gp_postid") int gp_postid)	GroupPurchaseServiceImpl.getGpPostByPostId(int gp_postid) BookServiceimpl.getBookId(int book_id) GroupPurchaseServiceImpl.getWhoApplierList(int gp_postid) GroupPurchaseServiceImpl.Detail_GroupPurchase(int gp_postid)	case1 (수정실패, 신청자1명이상) : ("Appliers", List<GpApply>) ("message", "이미 신청자가 있으므로 수정이 불가능합니다.") ("gpDetail", GroupPurchase) ("Thisbook", Book) case2(수정가능, 신청자0명) : ("Appliers", List<GpApply>) ("gpUpdate", GroupPurchase) ("Thisbook", Book)	case1 (수정실패, 신청자1명이상) : /groupPurchase/gp_detail_reader.jsp case2(수정가능, 신청자0명) : /groupPurchase/gp_update.jsp	forwarding	
공동구매 모집글 입력하기 위한 페이지(폼)로 이동	/groupPurchase/gp_list.jsp /groupPurchase/gp_list_reader.jsp /groupPurchase/gp_list_applier.jsp	/groupPurchase/uploadForm.do	-	GroupPurchaseReaderController.updateForm(HttpServletRequest request)	(본인작성) 공동구매 모집글 수정	/groupPurchase/gp_update.jsp	/groupPurchase/update.do	int book_id, int gp_postid, String post_title, Date gp_date, int limit, int dis_price, String post_contents, String account	GroupPurchaseReaderController.updateForm(HttpServletRequest request, @RequestParam("book_id") int book_id, @RequestParam("gp_postid") int gp_postid, @RequestParam("post_title") String post_title, @RequestParam("gp_date") Date gp_date)	BookServiceimpl.getBookId(int book_id) GroupPurchaseServiceImpl.Detail_GroupPurchase(int gp_postid) GroupPurchaseServiceImpl.Update_GroupPurchase(GroupPurchase Post) // new GroupPurchase(int gp_postid, post_contents, gp_date, limit, dis_price, gp_postid)	BookDao.getBookId(int book_id) GroupPurchaseDao.Detail_GroupPurchase(int gp_postid) GroupPurchaseDao.Update_GroupPurchase(GroupPurchase Post)	("gpDetail", GroupPurchase) ("Thisbook", Book)	/groupPurchase/gp_detail_reader.jsp	forwarding
공동구매 모집게시글 등록입력 > 등록하기	/groupPurchase/gp_upload.jsp	/groupPurchase/upload.do	int book_id, String book_name, String book_writer, String book_publisher, int book_price, String book_summary, String readerId, int readerMemberId, String post_title, Date gp_date, int dis_price, int limit, String post_contents, String account, MultipartFile img_file	GroupPurchaseReaderController.updateForm(HttpServletRequest request, @RequestParam("book_id") int book_id, @RequestParam("book_name") String book_name, @RequestParam("book_writer") String book_writer, @RequestParam("book_publisher") String book_publisher, @RequestParam("book_price") int book_price, @RequestParam("book_summary") String book_summary, @RequestParam("readerId") String readerId,	공동구매 참여신청	/groupPurchase/gp_detail.jsp	/groupPurchase/apply.do	int gp_postid	GroupPurchaseApplierController.apply(HttpServletRequest request, @RequestParam("gp_postid") int gp_postid)	GroupPurchaseServiceImpl.insertGpApply(GpApply newApply) GroupPurchaseServiceImpl.getGpPostByPostId(int gp_postid) GroupPurchaseServiceImpl.update(int gp_postid) // if limit==now	GroupPurchaseDao.Apply_GroupPurchase(int gp_postid) GroupPurchaseDao.insertGpApply(GpApply newApply) GroupPurchaseDao.getGpPostByPostId(int gp_postid) GroupPurchaseDao.update(int gp_postid) // if limit==now	redirect/groupPurchase/mApplier/list.do	redirecting	
				공동구매 신청취소	/groupPurchase/gp_detail_applier.jsp	/groupPurchase/cancel.do	int gp_postid	GroupPurchaseApplierController.cancel(HttpServletRequest request, @RequestParam("gp_postid") int gp_postid)	GroupPurchaseServiceImpl.Cancel_GroupPurchase(int gp_postid) GroupPurchaseServiceImpl.deleteGpApply(GpApply delApply)	GroupPurchaseDao.Cancel_GroupPurchase(int gp_postid) GroupPurchaseDao.deleteGpApply(GpApply delApply)	redirect/groupPurchase/mApplier/list.do	redirecting		
(본인작성)공동구매 모집글 삭제	/groupPurchase/gp_detail_reader.jsp	/groupPurchase/delete.do	int gp_postid	GroupPurchaseReaderController.deletePost(HttpServletRequest request, @RequestParam("gp_postid") int gp_postid)	(int gp_postid) BookServiceimpl.getBookId(int book_id) GroupPurchaseServiceImpl.getWhoApplierList(int gp_postid) GroupPurchaseServiceImpl.Detail_GroupPurchase(int gp_postid) GroupPurchaseServiceImpl.Delete_GroupPurchase(int gp_postid) BookServiceimpl.deletebook(int book_id)	BookServiceimpl.getBookId(int book_id) GroupPurchaseServiceImpl.getWhoApplierList(int gp_postid) GroupPurchaseServiceImpl.Detail_GroupPurchase(int gp_postid) GroupPurchaseServiceImpl.Delete_GroupPurchase(int gp_postid) BookServiceimpl.deletebook(int book_id)	case1 (삭제실패, 신청자1명이상) : ("Appliers", List<GpApply>) ("message", "이미 신청자가 있으므로 삭제가 불가능합니다.") ("gpDetail", GroupPurchase) ("Thisbook", Book) //case2는 redirect임	case1 (삭제실패, 신청자1명이상) : /groupPurchase/gp_detail_reader.jsp case2 (삭제성공, 신청자0명) : redirect/groupPurchase/mReader/list.do	forwarding/redirecting					

Part 3

기능별 Request 처리 흐름

경매 request
처리 흐름

경매 리스트 페이지로 이동	메인 페이지	auction	-	AuctionController.getList()	AuctionService.selectAuctionItemList()	AuctionDao.selectAuctionItemList	("list", List<AuctionItem>)	/auction/list.jsp	forwarding
경매 검색 (도서 이름, 작가 이름)	auction/list.jsp	auction/search.do	String key, String type	AuctionController.searchAuctionItem(key, type)	AuctionService.searchAuctionByItemName(key)	String searchAuctionByItemName(key) String searchAuctionByAuthor(String)	("list", List<AuctionItem>)	auction/auction_list_search.jsp	forwarding
내가 쓴 경매글 목록 보기	auction/list.jsp	auction/auctioneer.do	-	AuctionController.getAuctioneerList	AuctionService.selectAuctionItemByMemberId(memberId)	AuctionDao.selectAuctionItemByMemberId(memberId)	("list", List<AuctionItem>)	auction/auction_list_auctioneer.jsp	forwarding
내 입찰 목록 보기	auction/list.jsp	auction/bidder.do	-	AuctionController.getBidderList	AuctionService.selectAuctionBidByMemberId(memberId)	AuctionDao.selectAuctionBidByMemberId(memberId)	("list", List<AuctionItem>)	auction/auction_list_bidder.jsp	forwarding
경매 게시물 입력 장 이동	auction/list.jsp	auction/upload	-	-	-	-	-	auction/auction_upload.jsp	forwarding
경매 게시물 입력	auction/upload.jsp	auction/upload.do	AuctionItem ai, MultipartFile img_file	AuctionPostController.uploadPost(AuctionItem ai, MultipartFile img_file)	AuctionService.insertAuctionItem(AuctionItem ai)	AuctionDao.insertAuctionItem(AuctionItem ai)	("ai", AuctionItem)	/auction/auction_read.jsp	forwarding
경매 게시물 자세히 확인하기	auction/list.jsp	auction/item.do	au_itemId	AuctionPostController.getItem(String item_id)	AuctionService.selectAuctionItem(itemId)	AuctionDao.selectAuctionItem(itemId)	("ai", AuctionItem)	/auction/auction_read.jsp	forwarding
경매 게시물 삭제하기	/auction/auction_read.jsp	auction/delete.do	au_itemId	AuctionPostController.delete(String item_id)	AuctionService.deleteAuctionBid, AuctionService.deleteAuctionItem	AuctionDao.deleteAuctionBid, AuctionDao.deleteAuctionItem	-	auction/auction_delete_success	forwarding

경매	경매 게시물 수정하기로 이동	/auction/auction_read.jsp	auction/updatePost.do	au_itemId	AuctionPostUpdateController.updateForm	AuctionService.selectAuctionItem	AuctionDao.selectAuctionItem	("ai", AuctionItem)	/auction/auction_update.jsp	forwarding
	게시글 수정하기	auction/auction_update.jsp	auction/update.do	AuctionItem ai, MultipartFile	AuctionPostUpdateController.updatePost	AuctionService.updateAuctionItem(AuctionItem ai)	AuctionDao.updateAuctionItem(AuctionItem ai)	("ai", AuctionItem)	/auction/auction_read.jsp	forwarding
	경매 신청장 이동	auction/auction_read.jsp	auction/apply	au_itemId, startprice	-	-	-	-	auction/auction_bidApply.jsp	forwarding
	경매 신청하기	auction/auction_bidApply.jsp	auction/apply.do	AuctionBid ab	AuctionBidController.applyBid	AuctionService.insertAuctionBid(AuctionBid ab)	AuctionDao.insertAuctionBid(AuctionBid ab)	("auctionbid", bid)	auction/auction_bidComplete.jsp	forwarding
	경매 마감하기(낙찰하기) 및 낙찰자 정보 확인	auction/auction_read.jsp	auction/close	au_itemId	AuctionBidController.closeAuction	1) AuctionService.updateAuctionDealStatus(AuctionItem ai, int dealStatus) 2) AuctionService.updateAuctionBidStatus(AuctionItem ai, int bidStatus) 3) AuctionService.selectMemberByAuction(int memberId) 4) AuctionService.insertNoticeMessageByAuction(int memberId) # 알림장에 메시지 삽입	1) AuctionDao.updateAuctionDealStatus(AuctionItem ai, int dealStatus) 2) AuctionDao.updateAuctionBidStatus(AuctionBid ai, int bidStatus) 3) AuctionDao.selectMemberByAuction(int memberId) 4) AuctionDao.insertNoticeMessageByAuction(int memberId)	("member", member) ("ai", ai), ("ab", maxBid(낙찰가))	auction/auction_itemClose.jsp	forwarding

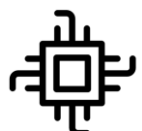
Part 4



구현 방법

Part 4

구현방법 - 사용기술



사용
기술



사용
언어



형상
관리

- 1. 모든 기능에 사용
- 2. com.bookchigo.
dao 패키지 내의 모든 파일들에 적용

Spring MVC (+ JSP)
Apache Tomcat 9
MyBatis Framework
Node JS, Socket IO

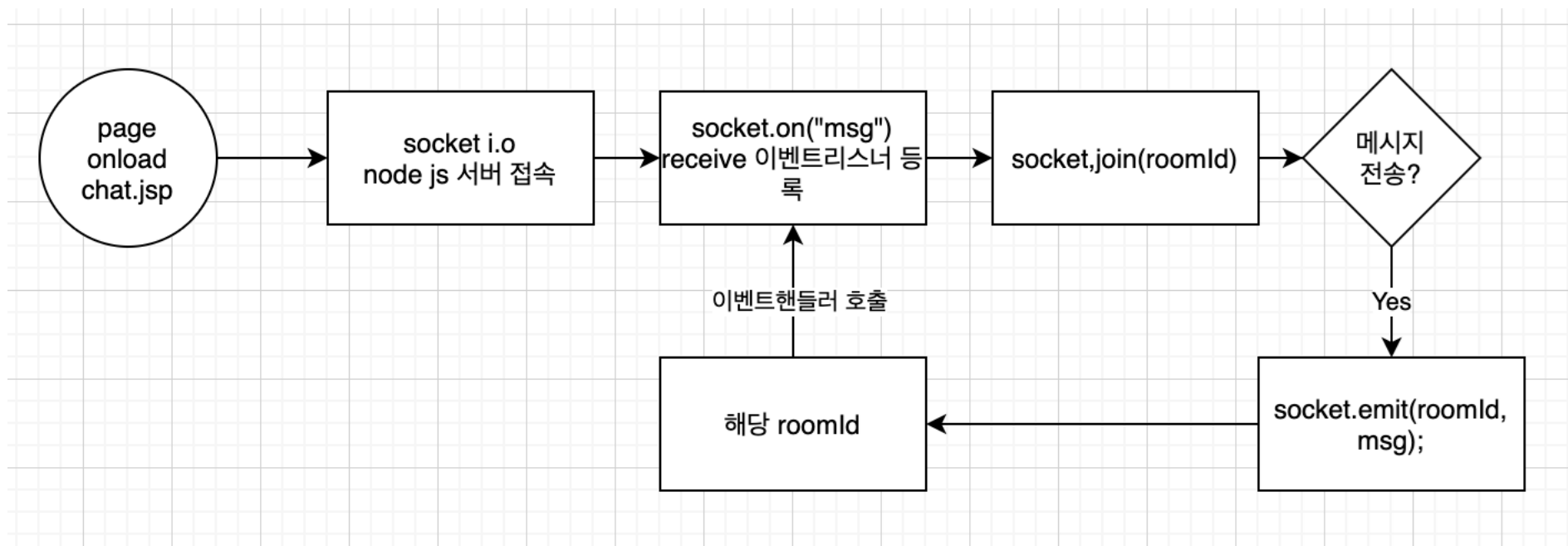
JAVA (JDK 1.8)
JSP
Javascript
HTML5
CSS

Github
Google Docs

- 1. 채팅 기능에 사용
- 2. 사용 파일
 - 1) Jsp 파일 :
chat/list.jsp,
chat/chat.jsp
 - 2) ChatController

Part 4

구현방법 - 채팅 구현



Part 4

구현방법 - 문제해결 사례



DB Lock

함께 프로젝트 작업을 하던 도중,
하나의 Table에 여러 클라이언트가
쿼리를 수행하여 데이터가 Update가
되지 않는 문제가 있었습니다.

oracle DB 에서 Session을 모두 끊어
문제를 해결하였습니다.



CSS Resource

JSP에서 Spring MVC framework
환경에서 JSP 페이지 안에서
JavaScript, CSS, 이미지 등
리소스파일을 포함하기 위하여

DispatcherServlet.xml 에
mvc:resource 구문을 추가하였습니다.



Image Upload

서비스에서 이미지 업로드를
구현하기 위하여

MultipartFile 로 파일을 전달하도록
하여, File을 서버에 작성하고 path를
DB에 데이터로 저장하였습니다.

그리고 View 다에서 해당 filepath를
img 태그로 렌더링 하였습니다.

Part 4

구현방법 - 프로젝트 파일 목록

```
java
├── com.bookchigo
│   └── controller
│       ├── AuctionBidController
│       ├── AuctionController
│       ├── AuctionPostController
│       ├── AuctionPostUpdateController
│       ├── ChatController
│       ├── GroupPurchaseApplierController
│       ├── GroupPurchaseController
│       ├── GroupPurchaseListController
│       ├── GroupPurchaseReaderController
│       ├── MainController
│       ├── MemberController
│       ├── NoticeController
│       ├── SellController
│       ├── SellStatusController
│       ├── SellUpdateController
│       ├── StatusForm
│       ├── StudyController
│       ├── UpdateForm
│       └── WishController
```

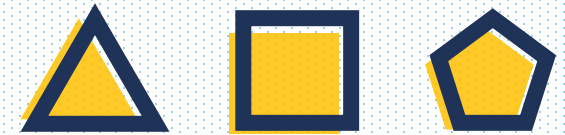
```
dao
├── mybatis
│   ├── AuctionDao
│   ├── BookDao
│   ├── ChatDao
│   ├── GroupPurchaseDao
│   ├── MemberDao
│   ├── NoticeDao
│   ├── SellDao
│   └── StudyDao
└── domain
    ├── AuctionBid
    ├── AuctionItem
    ├── Book
    ├── BuyFin
    ├── Chat
    ├── ChatMessage
    ├── ChatRoom
    ├── GpApply
    ├── GroupPurchase
    ├── Member
    ├── Notice
    ├── SellItem
    ├── Study
    └── Wishlist
```

```
service
├── AuctionService
├── AuctionServiceImpl
├── BookService
├── BookServiceImpl
├── ChatService
├── ChatServiceImpl
├── GroupPurchaseService
├── GroupPurchaseServiceImpl
├── MemberService
├── MemberServiceImpl
├── NoticeService
├── NoticeServiceImpl
├── SellServiceFacade
├── SellServiceImpl
├── StudyService
├── StudyServiceImpl
└── util
    └── ImageUtil
```

```
view
├── auction
│   ├── auction_bidApply.jsp
│   ├── auction_bidComplete.jsp
│   ├── auction_delete_success.jsp
│   ├── auction_itemClose.jsp
│   ├── auction_list.jsp
│   ├── auction_list_auctioneer.jsp
│   ├── auction_list_bidder.jsp
│   ├── auction_list_search.jsp
│   ├── auction_read.jsp
│   ├── auction_update.jsp
│   └── auction_upload.jsp
├── chat
│   ├── chat.jsp
│   └── list.jsp
└── groupPurchase
    ├── gp_detail.jsp
    ├── gp_detail_applier.jsp
    ├── gp_detail_finish.jsp
    ├── gp_detail_finish_reader.jsp
    ├── gp_detail_reader.jsp
    ├── gp_list.jsp
    ├── gp_list_applier.jsp
    ├── gp_list_reader.jsp
    ├── gp_update.jsp
    ├── gp_upload.jsp
    └── ReadMe.jsp
```

```
main
├── IncludeBottom.jsp
├── IncludeTop.jsp
├── IncludeTopMain.jsp
├── main.jsp
├── member
│   ├── login.jsp
│   ├── my.jsp
│   └── register.jsp
├── notice
│   ├── notice_list.jsp
│   └── notice_post.jsp
├── sell
│   ├── sell_finish.jsp
│   ├── sell_history.jsp
│   ├── sell_list.jsp
│   ├── sell_publisher.jsp
│   ├── sell_seller.jsp
│   ├── sell_status.jsp
│   ├── sell_update.jsp
│   ├── sell_upload.jsp
│   └── sell_wishlist.jsp
└── study
    ├── enroll.jsp
    ├── list.jsp
    ├── dispatcher-servlet.xml
    └── web.xml
```

Part 5



팀원 업무 분담 및 결론과 소감

Part 5

팀원 업무 분담 내용

이름	역할 - 기능을 위주로 업무 분담
김도연	중고 판매 관련 기능
나하윤	공동구매와 관련된 기능
박지우	채팅 및 회원 관련 기능
성지운	중고 책 경매 관련 기능

- 주요 설계는 다같이 진행
- 기능 위주로 업무를 분담하긴 했지만 통합을 진행하면서 같이 구현을 한 부분도 多

‘북치고 중고치고’는

대학생들이 사용하던 중고책을 판매할 수 있는 거래 플랫폼입니다.

중고 책 판매 기능은 물론이며, 기존의 기능들과는 차별화된 특별한 기능들이 몇 가지 추가되어 있습니다.

★ 첫째, 경매기능, 입찰자는 입찰을 신청할 수 있고, 판매자는 경매를 마감할 수 있습니다.

★ 둘째, 스터디 그룹 기능, 스터디 호스트는 특정 책을 가지고 스터디원을 모집할 수 있습니다.

★ 셋째, 채팅 기능, 판매자와 구매자간 채팅, 스터디원 끼리 단체 채팅 기능을 지원합니다.

★ 넷째, 도서 공동구매기능, 사용자는 원하는 도서를 다른 사용자와 공동구매를 할 수 있도록 지원합니다.

이외에도 로그인 및 회원가입 기능, 개인 정보 관리 및 탈퇴 기능, 즐겨찾기 기능과 같은 많은 기능이 존재합니다.

‘북치고 중고치고?’는 MVC 구조를 활용하여 다양한 컨트롤러와 모델, 그리고 뷰를 사용하여 개발하였습니다.

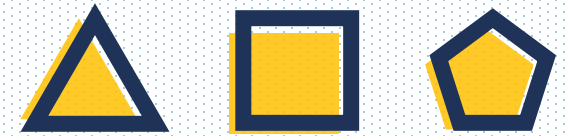
궁극적으로 이 웹 페이지는 사용자들끼리 중고책을 거래할 수 있는 더 나은 환경을 제공하기 위하여 개발되었습니다.

이번 프로젝트를 통해서 Spring MVC의 동작 방식과 프로젝트를 설계하는 방법에 대하여 배울 수 있었습니다.
MyBatis를 사용하여 RDB 와 연동하는 방법을 알고, Domain 클래스와 매핑하여 Model 단계에 데이터를 담아 View 까지 전달하는 과정을 개발 과정에서 몸소 실습하고 확인할 수 있었습니다.

뿐만 아니라, 팀원들과 협업을 통하여 협업 프로세스를 배우고 서로 간의 데이터 프로토콜을 맞추며
더 나은 방법으로 프로젝트를 진행하기 위해 노력하였습니다.

이후에는, 지금 현재 사용한 기술에서 더 나아가 수업시간을 통해서 학습한 Spring Boot를 통해 Xml의 의존성을 줄이고,
JPA 및 Hibernate를 사용하여 쿼리 의존성을 낮추고, JSP 대신 대표적인 템플릿 언어인 Mustache나 Thymeleaf 를
사용해볼 수도 있을 것 같습니다.

2분반 8팀 : 책사조
박지우 김도연 나하윤 성지운



감사합니다!