

SUMMER PetStore

소프트웨어 시스템 개발 01분반 9조 쌤머

20170962 이예린

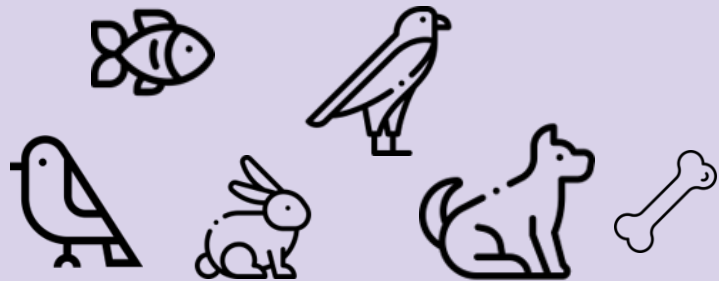
20160965 성연주

20170989 한수현

20151044 조현아

목차

1. 개요 및 특징
2. 세부 기능
3. 시스템 구조 (구성 요소)
4. 구현 방법 (사용기술, 알고리즘, 문제해결 방법 등)
5. 업무 분담 (세부적인 분담 내용)
6. 결론



1. 개요 및 특징

- summerpetstore는 반려동물의 물품을 판매하고 구매하는 사이트

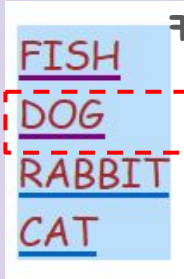
〈시스템 개요〉

- 일반물품 구매
- 장터물품 구매,등록
- 경매
- 장바구니
- 사용자 관리

〈시스템 특징〉

- Spring framework기반 개발
- java, html, jsp 사용
- Spring MVC 모델 사용
- DB연계(SQL Developer 이용)
- My Batis 사용

2. 세부 기능 - 일반물품



카테고리별 물품 보기

Auction * Group Purchasing * Market			
Products for sale at Spet Store			
Item ID	Category	Name	Price
20001	dog	SPET puppy sweets	20000won
10001	dog	SPET pup shampoo	9000won
10002	dog	SPET Simply Nourish	9000won
10003	dog	SPET Canidae	25000won
10004	dog	SPET Spiky Ball	5000won

Products for sale at Market			
Item ID	Category	Name	Price
20011	dog	Strawberry Dry Shampoo	9000won
20012	dog	Apple Dry Shampoo	9000won
20013	dog	Kiwi Dry Shampoo	9000won
20002	dog	Grreat Choice Dog Carrier	60000won

Products for Auction			
Auction Number	Name	D-DAY	Current Price
30060	test97	183	0won
30029	My special pad	5	50000won
30040	Pet Cartpet Shampoo	10	10000won

일반 상품 자세히보기

Auction * Group Purchasing * Market

SPET puppy sweets

Product Name : SPET puppy sweets
 Pet Kind : dog
 Item Kind: feed
 Price : 20000
 Description : worm jelly
 Item Status : in stock

[Add Cart](#)

일반물품 장바구니 담기

My Cart

SUMMER PET STORE

Item Number:20001
20000Won

Quantity :

[update](#)
[Remove](#)

2. 세부 기능 - 장터

Register Market

마켓 등록

Product Name :

Pet Kind :

Item Kind :

Price :

Description :

register

cancel

마켓 등록 완료

Congratulation!! You success to register new item in the Summerpetstore Market!

view market item detail:

- Product Name : test20160965
- Pet Kind : dog
- Item Kind: feed
- Price : 9000
- Description : test

[Go to Market List](#)

[item pic]

Price: 9000

test20160965

Item Number: 20041

Status:

Price: 9000

마켓 상품 자세히 보기

test20160965

Product Name : test20160965

Pet Kind : dog

Item Kind: feed

Price : 9000

Description : test

Item Status : in progress

[Add Cart](#)

마켓 상품 장바구니 추가

SUMMER
PET STORE

Quantity :

Item Number:20041
9000Won

update

Remove

*카테고리별 물품 보기에서도 장터 물품 확인 가능

2. 세부 기능 - 경매

Register Auction

경매 등록

Title :

Pet Kind : --pet kind-- ▼

Deadline :

Description :

Minimum Price :

Register

경매 등록 완료

test20160965

D-Day 183

Deadline 2021-01-01 00:00:00.0

Auction Number: 30061

Auction Name : test20160965

Pet Kind : cat

Current Max Price: 0

User :

Description : test

Minimum Price : 0

Seller : yerinie

Status : in progress

Register Price

단, 현재 1등 입찰자의 가격보다 낮으면 입찰 불가

Current Price: 0 by User:

×

입찰가 입력

Congratulation!! You success to register new auction in the Summerpetstore Auction!

[Go to Auction List](#)

입찰 성공

[item pic]

Auction Number: 30061

Status: in progress

Minimum Price: 0

Current Price: 0

Auction Name: [test20160965](#)

경매 상품 자세히보기

D-183

Current Price: 0

Auction Name: [test20160965](#)

Auction Number: 30061

Status: in progress

Minimum Price: 0

Current Price:

*카테고리별 물품 보기에서도 경매 물품 확인 가능

2. 세부 기능 - 장바구니

SUMMER
PET STORE.®

Item Number:20010
8000Won

수량 변경 버튼

update
Remove

Quantity : 1

SUMMER
PET STORE.®

Item Number:20001
20000Won

해당 장바구니 물품 삭제

update
Remove

Quantity : 1

Total price : 28000 won
submit

Payment Details

Card Type: --Card Type--
Card Number:
Expiry Date (MM/YYYY):
CVC Number:
Zip Code:

Billing Address

Name: yerin lee
Address: Seoul, Korea
Phone Number: 010-6666-6644

ok

주문서 작성

주문 완료

You success to order products!

[go to home](#)

*주문시 장바구니 비워짐

2. 세부 기능 - 사용자 관리

*로그인/로그아웃/회원가입은 생략함

*로그인/로그아웃할 경우, 상단 로그인/아웃 버튼 변경됨

[Edit My Info](#)

My Auction

My Group Purchases

My Market

SUMMER
PET STORE

사용자 정보 수정

일반 사용자 MyPage(등록한 장터 목록 확인)

20042

in progress

9700

내가 등록한 장터 삭제 버튼

[Delete Item](#)

User Information

User ID: yeonju97

password:

Name: yeonju seong

Email: yj@naver.com

Address: seoul

Phone: 01033712849

Save Account Information

[WITHDRAWAL](#)

탈퇴 버튼

[Edit My Info](#)

My Auction

[new Auction](#)

[item pic]

[test97](#)

[item pic]

My Group Purchases

My Market

SUMMER
PET STORE

Admin 사용자 MyPage(등록한 경매/장터 목록 확인)

30044

in progress

0

500

30060

in progress

0

0

20001

in stock

20000

[Delete Item](#)

2. 세부 기능 - 프로젝트 처리흐름도

	세부기능	input view	request URL	Controller
사용자관리	회원가입	/spetstore/user/register.jsp	/spetstore/user/newAccount.do	AccountFormController
	회원탈퇴	/spetstore/user/myPage/editAccount.jsp	/spetstore/index	UserController
	로그인	/spetstore/user/SignonForm.jsp	/spetstore/signon.do	UserController
	장바구니 물품 삭제	/spetstore/user/myPage/cart.jsp	/spetstore/user/myPage/cart/remove	CartController
	장바구니 물품 결제	/spetstore/user/PayForm.jsp	/spetstore/user/order/PaidSuc.do	CartController
	회원정보수정	/spetstore/user/editAccount.jsp	/spetstore/user/editAccount.do	AccountFormController
	내놓은 물건 보기(장터)	/spetstore/user/myPage.jsp	/spetstore/market/detail	MarketController
일반상품	업로드한 경매 보기(Admin only)	/spetstore/user/summerPage.jsp	/spetstore/auction/detail	AuctionController
	장바구니담기	/spetstore/spetitem/sListDetail.jsp	/spetstore/market/addCart/{itemid}	CartController
	일반구매물품자세히보기	/spetstore/spetitem/categoryView.jsp	/spetstore/spetitem/detail	ShowDetailController
	장터물품등록	/spetstore/market/mRegister.jsp	/spetstore/market/mSearch/mRegisterSuc.do	MarketController
장터	장터물품자세히보기	/spetstore/market/mSearch.jsp	/spetstore/market/detail	MarketController
	장터물품삭제	/spetstore/user/myPage.jsp	/spetstore/market/delete/{itemid}	MarketController
	장바구니담기	/spetstore/market/mDetail.jsp	/spetstore/market/addCart/{itemid}	CartController
	경매정보자세히보기	/spetstore/auction/aSearch.jsp	/spetstore/auction/detail	AuctionController
경매	입찰	/spetstore/auction/priceRegister.jsp	/spetstore/auction/bid/{curMaxPrice}/{auctionId}	BidController
	경매등록	/spetstore/auction/mRegister.jsp	/spetstore/auction/aSearch/aRegisterSuc.do	AuctionController
기타	카테고리별 정보보기	spetstore/index.jsp	/spetstore/category/{cname}	ShowCategoryViewController

Controller's method & Parameter	Service	DAO	result view
on Submit(HttpServletRequest, HttpSession, @ModelAttribute("user"), BindingResult)	insertUser(account)	UserDAO.insertUser(account)	/spetstore/user/sucRegister.jsp
deleteUser(@RequestParam("userId"), HttpSession, Session Status)	deleteUser(userId)	UserDAO.deleteUser(userId)	/spetstore/index.jsp
handleRequest(HttpServletRequest, @ModelAttribute("currentLogin"), Model)	getUserByid(userId, password)	UserDAO.getUserByid(userId, password)	/spetstore/index.jsp
deleteItem(@RequestParam("itemId"), HttpServletRequest)	showCartList(userId)	CartDAO.showCartList(userId)	/spetstore/user/myPage/Cart.jsp
finishOrder(@ModelAttribute("newOrder"), @ModelAttribute("orderForm"), Model, HttpServletRequest)	createOrder(orderForm)	CartDAO.createOrder(orderForm)	/spetstore/user/Paid Suc.jsp
on Submit(HttpServletRequest, HttpSession, @ModelAttribute("user"), BindingResult)	updateUser(account)	UserDAO.updateUser(account)	/spetstore/user/myPage.jsp
showMarketDetail(@RequestParam("itemId"))	showInfo(itemId)	MarketDAO.showInfo(itemId)	/spetstore/market/mDetail.jsp
showAuctionDetail(@RequestParam("auctionId"), HttpServletRequest, Model)	showInfo(auctionId)	AuctionDAO.showInfo(auctionId)	/spetstore/auction/aListDetail.jsp
addCartSpetitem(@PathVariable itemId, HttpServletRequest)	addCart(userId, itemId), showInfo(itemId)	CartDAO.addCart(userId, itemId), MarketDAO.showInfo(itemId)	/spetstore/spetitem/sListDetail.jsp
showSpetitem(@RequestParam("itemId"), Model)	showdetailInfoSP(itemId)	SProuctDAO.showdetailInfoSP(itemId)	/spetstore/spetitem/sListDetail.jsp
registerMarket(@ModelAttribute("marketForm"), Session Status, Model, HttpServletRequest)	registerItem(marketForm)	MarketDAO.registerItem(marketForm)	/spetstore/market/mRegisterSuc.jsp
showMarketDetail(@RequestParam("itemId"))	showInfo(itemId)	MarketDAO.showInfo(itemId)	/spetstore/market/mDetail.jsp
deleteMarket(@PathVariable int itemId, Model)	deleteItem(itemId)	MarketDAO.deleteItem(itemId)	/spetstore/market/mSearch.jsp
addCartMarket(@PathVariable itemId, HttpServletRequest)	addCart(userId, itemId), showInfo(itemId)	CartDAO.addCart(userId, itemId), MarketDAO.showInfo(itemId)	/spetstore/market/mDetail.jsp
showAuctionDetail(@RequestParam("auctionId"), HttpServletRequest, Model)	showInfo(auctionId)	AuctionDAO.showInfo(auctionId)	/spetstore/auction/aListDetail.jsp
bid(@RequestParam("bidPrice"), @PathVariable curMaxPrice, @PathVariable auctionId, HttpServletRequest)	bid(userId, myPrice, auctionId)	AuctionDAO.bid(userId, myPrice, auctionId)	/spetstore/auction/aRegisterSuc.jsp
registerAuction(@ModelAttribute("auctionForm"), HttpServletRequest, Model)	registerAuction(auctionForm)	AuctionDAO.registerAuction(auctionForm)	/spetstore/auction/aRegisterSuc.jsp
showCateView(@PathVariable("cname"), Model)	showInfoSP(cname), showInfoMP(cname), showInfoAP(cname)	showInfoSP(cname), MarketDAO.showInfoMP(cname), AuctionDAO.showInfoAP(cname)	/spetstore/spetitem/categoryView.jsp

단, '장바구니, 장터 물품 등록, 경매 등록, 장바구니 담기'의 경우에는 로그인이 안되어있는 경우, 로그인 창으로 이동함

스프레드 시트 링크 : https://docs.google.com/spreadsheets/d/1FNJ-M0-g12FB6JY4e8dkrS_jii8-K8DSb9SDkNNQafMA/edit?usp=sharing

3. 시스템 구조

클라이언트

1 처리요청(URL)

Controller

AccountFormController.java
AuctionController.java
BidController.java
CartController.java
MarketController.java
MoveToMRegisterController.java
ShowCategoryViewController.java
ShowDeatilController.java
UserController.java
...

2 Request 처리 요청

3 Request 처리

Model

AuctionService.java-AuctionDAO.java
-AuctionMapper.java
CartService.java-
CartDAO.java-CartMapper.java
MarketService.java-MarketDAO.java-
MarktetMapper.java
...

4 ModelAndView
return

Model data를
전달받아 Response
생성(jsp 이용)

5

View

/user/editAccount.jsp
/user/myPage/Cart.jsp
/spetitem/categoryView.jsp
/market/mDetail.jsp
/auction/aListDetail.jsp
/auction/aRegister.jsp
...

4. 구현 기능 - 문제 해결방법

1) jar파일 부재로 발생한 오류

메시지 Request processing failed; nested exception is org.mybatis.spring.MyBatisSystemException: nested exception is org.apache.ibatis.exceptions.PersistenceException:

-> WEB-INF/lib에 jar 파일 등록

2) Controller 內 RequestMapping 정보 오류

HTTP 상태 404 – 찾을 수 없음

타입 상태 보고

설명 Origin 서버가 대상 리소스를 위한 현재의 representation을 찾지 못했거나, 그것이 존재하는지를 밝히려 하지 않습니다.

-> request url 올바르게 수정

3) 리스너 설정 오류

HTTP 상태 500 – 내부 서버 오류

타입 예외 보고

메시지 서블릿 [dispatcherServlet]을(를) 위한 Servlet.init() 호출이 예외를 발생시켰습니다.

-> 설정 파일(web.xml) 올바르게 수정

4. 구현 기능 - 문제 해결방법

4) Controller, Service 등 경로 오류

```
<context:component-scan base-package="dongduk.cs.ssd.summerpetstore.controller" />
```

-> web.xml 에서 '`<listener-class>org.springframework.web.context.ContextLoaderListener</listener-class>`'

부분 주석을 처리를 취소함

-> 주석 처리를 했던 이유: 주석 처리를 하지 않으면 첫 화면이 등장하지 않고 클래스 리스너 오류가 난다.

->리스너 오류가 났던 이유: jdbc.oracle.properties를 포함하지 않았음 -> 이 파일을 포함 하여 위의 모든 오류 해결

4. 구현 기능 - 사용기술

1) Validation 기능

: controller에서 validator를 이용해
회원가입, 회원수정에서 입력값 검증

User Information

User ID:	<input type="text"/>	User ID is required.
password:	<input type="text"/>	Password is required.
Name:	<input type="text"/>	Name is required.
Email:	<input type="text"/>	Email address is required.
Address:	<input type="text"/>	Address is required.
Phone:	<input type="text"/>	Phone number is required.

[Save Account Information](#)

Passwords did not match. Matching passwords are required.

User Information

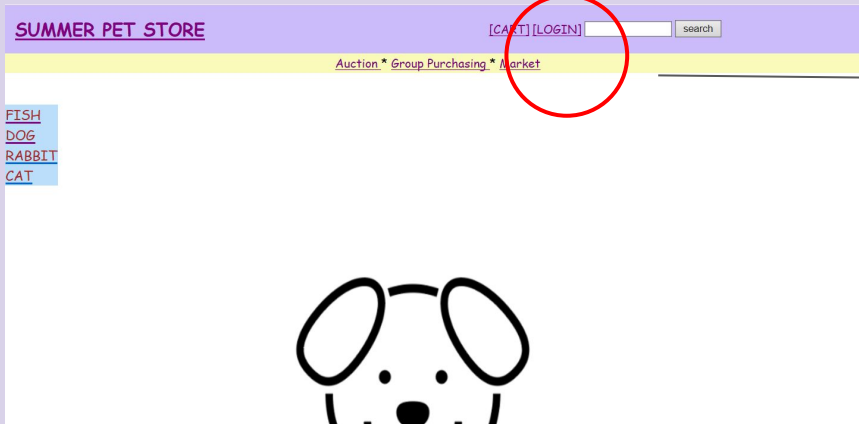
User ID:	sh
password:	<input type="text"/>
Name:	sh222
Email:	sh@naver.com
Address:	seoul
Phone:	010-222-2222

[Save Account Information](#)

[WITHDRAWAL](#)

2) interceptor기능

: 로그인을 하지 않은 상태에서 장바구니(cart)로 간다면 interceptor를 통해 로그인 창으로 이동



[Auction](#) * [Group Purchasing](#) * [Market](#)

Please enter your username and password.

Username:	<input type="text"/>
Password:	<input type="password"/>

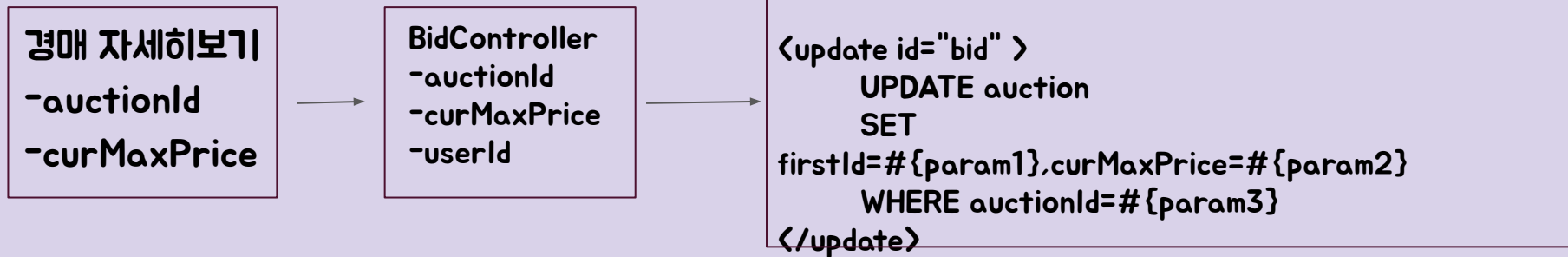
[login](#)

[register now](#)

4. 구현 기능 - 알고리즘

1) 경매 입찰:

- 경매 자세히 보기 화면에는 경매 번호와, 현재 최고가를 포함한 정보가 있다.
- BidController에서 PathVariable로 받아온 현재 최고가와, 경매 번호를 받아온다.
- UserSession을 통하여 현재 로그인 되어있는 계정을
현재 최고가와 경매번호와 함께 Service와 Dao를 넘겨주어 최고가와 해당 유저정보를 수정한다
- 경매 성공화면으로 넘긴다.



4. 구현 기능 - 알고리즘

2) 장바구니 재고 확인:

- 이미 장바구니에 존재하는 물건인 경우와
장바구니에 존재하지 않은 경우으로 나눈다.(itemId, userId로 검사)
- 이미 존재하는 물건일 경우:
장바구니에 존재하는 quantity와 재고량과 비교하여
재고량이 더 클 경우 InStock 을 true(1)로 설정하고, 반대의 경우 false(0)로 설정한다.
그리고 quantity 에 + 1을 한다(update).
- 장바구니에 존재하지 않은 물건일 경우:
재고량이 1보다 클 경우 InStock 을 true(1)로 설정하고, 반대의 경우 false(0)로 설정.
카트에 정보추가(insert)

다음에 계속(코드)

4. 구현 기능 - 알고리즘

```
public void addCart(String userId, int itemId) {  
    if(cmapper.countItemId(userId, itemId) > 0) {  
        //이미 존재하는 물건일 경우  
        int q = cmapper.getQuantity(itemId, userId);  
        ItemModel im = cmapper.findItembyitemId(itemId);  
        if(im.getStock() < q) { //재고량 확인  
            cmapper.updateInStock(0, itemId, userId);  
        }  
        cmapper.incrementQuantity(userId, itemId); //quantity + 1  
    }  
    else { //존재하지 않은 물건일 경우  
        ItemModel im = cmapper.findItembyitemId(itemId);  
        int inStock;  
        int quantity = 1;  
        if(im.getStock() >= quantity) { //재고량 확인  
            inStock = 1;  
        }else {  
            inStock = 0;  
        }  
        cmapper.addCart(itemId, im.getName(), im.getPrice(), inStock, quantity, userId);  
    }  
}
```


5. 업무 분담

20160965 성연주

<장바구니>

- 삭제
- 장바구니 총액계산 및 업데이트
- 주문과 연결
- 장바구니 삭제

- 주문

<경매>

- 경매 정보보기

<장터>

- 장터 자세히보기
- 장터 정보보기

<기타>

- 모든 View기본 토대
+ 수정
- 화면 디자인

20170962 이예린

<경매>

- 경매 목록 보기
- 입찰
- 경매 등록
- 경매 자세히보기

<장터>

- 장터 추가
- 장바구니 추가

<일반(공식물품) 구매>

- 장바구니 추가

<사용자>

- 마이페이지에 해당 사용자 장터 넣기
- 로그인 인터셉터

<기타>

- db관리
- 회의록 정리
- View수정
- 모든 mapper 토대
- 화면 디자인

20170989 한수현

<일반(공식물품) 구매>

- 물품 자세히보기
- 카테고리별 물품보기

<사용자>

- 로그인
- 로그아웃
- 회원 정보수정
- 회원 탈퇴

-마이페이지에 해당사용자 경매 넣기

-마이페이지에서 올린 장터 삭제

<기타>

- db관리
- View수정
- 모든 Controller 토대
- 화면 디자인

6. 결론

이번 코로나 사태로 대면강의가 아닌 온라인 강의로 진행되어서 팀프로젝트하는데 전에 했던 것과 다른 방식이라 색달랐다. 실제로 만나서 회의하기보다 행아웃으로 온라인 회의를 진행하였으나, 팀원들이 잘 따라와서 큰 문제없이 진행되었다. 그러나 개발 시작 후 팀원 중 한명이 참여를 안하게 되어 계획에 차질이 생기고, 초반에 오류를 잡는데에 시간을 많이 써 시간이 촉박했음이 많이 아쉬웠다.

이번 summerpetstore를 진행하면서 처음으로 spring framework을 사용해봤는데 처음에는 생각보다 어렵고 복잡하게 느껴졌지만 수업을 듣고 공부하고 팀원들과 도와가면서 진행하다보니 점차 적응하고 이해하면서 진행했다. 다만 시간이 부족해서 처음에 계획했던 기능을 다 구현하지는 못해서 아쉽긴하지만 할 수 있는 최선을 다해서 후회는 없다. 한학기 동안 팀원들이 각자 자신이 맡은 바를 열심히 해서 이런 최선의 결과를 얻었다고 생각해 뜻 깊었던 프로젝트였다. 소프트웨어 시스템 개발 수업을 통한 spring 기반 팀 프로젝트를 하면서 많은 것을 배웠던 한학기였다.

도움을 주신 박창섭 교수님 감사합니다.