

# 편의점 온라인 통합 시스템 개발

팀 명 : sunflower

지도 교수 : 이병천 교수님

팀 장 : 최용준

팀 원 : 김병관

염태원

김민경

백수연

2019. 10.

중부대학교 정보보호학과

# 목 차

<b>1. 서론</b>	
1.1 프로젝트의 필요성 .....	2
1.2 프로젝트의 목표 .....	2
<b>2. 관련 연구</b>	
2.1 Java/Jsp .....	2
2.2 Tomcat8.5 .....	2
2.3 Mysql .....	3
2.4 Windows10 .....	3
2.5 Md5 .....	3
2.6 Port forwarding .....	3
<b>3. 본론</b>	
3.1 기능 설계 .....	4
3.2 DB 자료구조 설계 .....	4
3.2.1 전체 DB 테이블 .....	4
3.2.2 사용자(고객) 정보 DB .....	5
3.2.3 사용자(점주) 정보 DB .....	5
3.2.4 구매 내역 DB .....	5
3.2.5 주문 내역 DB .....	5
3.3 구현 상세 내용 .....	6
3.3.1 초기화면 .....	6
3.3.2 새로운 사용자 등록 .....	6
3.3.3 마이페이지 .....	8
3.3.4 커뮤니티 .....	10
3.3.5 새로운 편의점 등록 .....	12
3.3.6 새로운 물품 등록 .....	14
3.3.7 마일리지 적립 .....	16
3.3.8 물품구매 및 장바구니 .....	17

3.3.9	구매내역 확인 .....	19
3.3.10	비밀번호 해시 값으로 저장 .....	20
<b>4.</b>	<b>결론 및 향후계획 .....</b>	<b>21</b>
<b>5.</b>	<b>첨부</b>	
5.1	발표 자료 .....	22
5.2	소스코드 .....	36

# 1. 서론

## 1.1 프로젝트의 필요성

편의점 물품을 웹에서 구매할 수 있게끔 만들어 놓은 편의점 기업은 두 개 정도뿐이다. 그마저도 하나는 어플을 설치해야 한다. 그리고 웹에 상품을 구매할 수 있게 만든 두 개의 기업이 있다지만 웹에서 판매되는 상품들은 각 매장의 점주들이 가지고 있는 상품이 팔리는 것이 아니기 때문에 편의점 점주들 입장에서는 자기 자신의 편의점 홈페이지가 아닌 자신의 브랜드 홈페이지에서 판매가 이루어지는 것은 의미가 없다. 그렇기에 우리가 생각해낸 해결방안은 편의점 통합 웹 서비스를 만들어 점주가 자신의 편의점 위치와 상품을 등록·관리할 수 있게끔 만들게 된 것이 “너만의 편의점” 웹 서비스이다.

## 1.2 프로젝트의 목표

너만의 편의점 서비스를 만들게 되어 점주와 소비자에게 주는 장점은 시스템을 통해 소비자는 해당 편의점의 물품 재고를 미리 확인 및 주문 결제 구현할 수 있고 직관적인 시스템을 통해 소비자와 점주의 편의성 증대를 기대해볼 수 있다. 편의점 점주 입장에서는 효율적이고 투명한 편의점 관리가 가능하다는 것. 특히 시스템 보안 업그레이드에 중점을 두었다. 사업적인 추진전략으로는 소비자는 자신의 입맛대로 다양한 편의점의 상품과 후기를 확인 비교하여 효과적으로 소비할 수 있고 점주는 자신의 편의점을 알릴 수 있게 된다. 편의점 점주들은 자신의 편의점을 안전하게 알릴 수 있는 좋은 계기가 될 것이고 여러 방면으로 수익을 창출할 수 있는 방법을 생각하며 웹서비스를 구축하게 되었다.

# 2. 관련 연구

## 2.1 Java/Jsp

JSP는 HTML 내에 자바 코드를 삽입하여 웹 서버에서 동적으로 웹 페이지를 생성하여 웹 브라우저에 돌려주는 언어이다. Java EE 스펙 중 일부로 웹 애플리케이션 서버에서 동작한다. JSP는 실행 시에는 자바 서블릿으로 변환된 후 실행되므로 서블릿과 거의 유사하다고 볼 수 있다. 하지만, 서블릿과는 달리 HTML 표준에 따라 작성되므로 웹 디자인하기에 편리하다. 1999년 썬 마이크로시스템즈에 의해 배포되었으며 이와 비슷한 구조로 PHP, ASP, ASP.NET 등이 있다.

## 2.2 Tomcat 8.5

아파치 톰캣(Apache Tomcat)은 아파치 소프트웨어 재단에서 개발한 서블릿 컨테이너(또는 웹 컨테이너)만 있는 웹 애플리케이션 서버이다. 톰캣은 웹 서버와 연동하여 실행할 수 있는 자바 환경을 제공하여 자바 서버 페이지(JSP)와 자바 서블릿이 실행할 수 있는 환경을 제공하고 있다. 톰캣은 관리 툴을 통해 설정을 변경할 수 있지만, XML 파일을 편집하여 설정할 수도 있다. 그리고, 톰캣은 HTTP 서버도 자체 내장하기도 한다. 아파치

톰캣은 Apache Licence, Version 2를 채용한 오픈 소스 소프트웨어로서, 자바 서버 페이지이나 자바 서블릿을 실행하기 위한 서블릿 컨테이너를 제공하며, 상용 웹 애플리케이션 서버에서도 서블릿 컨테이너로 사용하는 경우가 많다. 버전 5.5 이후는 기본적으로 Java SE 5.0 이후를 대응한다.

### 2.3 Mysql

MySQL은 전 세계적으로 가장 널리 사용되고 있는 오픈 소스 데이터베이스이며, MySQL AB 사가 개발하여 배포/판매하고 있는 데이터베이스이다. 표준 데이터베이스 질의 언어 SQL(Structured Query Language)을 사용하는 개방 소스의 관계형 데이터베이스 관리 시스템(RDBMS)이며, 매우 빠르고 유연하며 사용하기 쉬운 특징이 있다. 다중사용자, 다중 스레드를 지원하고, C, C++, Eiffel, Java, Perl, PHP, Python 스크립트 등을 위한 응용프로그램 인터페이스를 제공한다. 또한 유닉스나 리눅스, Windows 운영체제 등에서 사용할 수 있다. 또한 MySQL은 리눅스 운영체제와 Apache 서버 프로그램, MySQL, PHP 스크립트 언어 구성은 상호 연동이 잘 되면서도 오픈소스로 개발되는 무료 프로그램이어서 홈페이지나 쇼핑몰 등 일반적인 웹 개발에 널리 사용된다.

### 2.4 Windows 10

Windows 10 은 Microsoft의 윈도우 계열의 개인용 컴퓨터 운영 체제이다. 2015년 7월 29일 일반 사용자에게 공개되었으며, Windows Vista 이후 Windows 7, 8로 버전이 넘버링 되어 왔는데, 9를 뛰어넘고 Windows 10으로 이름 붙였다. 이는 Microsoft가 모든 장치에서 포괄적으로 동작하는 다양한 플랫폼을 나타내기 위한 의도에 의한 것이다. 커널 또한 Windows Vista 이후 6.x 버전에서 바로 10.0으로 출시되었다.

### 2.5 md5

MD5(Message-Digest algorithm 5)는 128비트 암호화 해시 함수이다. RFC 1321로 지정되어 있으며, 주로 프로그램이나 파일이 원본 그대로인지를 확인하는 무결성 검사 등에 사용된다. 1991년에 로널드 라이 베스트가 예전에 쓰이던 MD4를 대체하기 위해 고안했다. 메시지의 무결성(Message Integrity) 점검을 위해 이용하는 하나의 알고리즘이다. 간이 전자 우편 전송 프로토콜(SMTP: Simple Mail Transfer Protocol) 서버 소프트웨어인 샌드메일(sendmail)이나 도메인 네임 서버(DNS)의 바인드(BIND) 소프트웨어 인증, 그리고 데이터베이스 등에서 널리 사용된다. 초기의 MD2가 8 비트 버전인데 비해 MD3나 MD4의 개선형인 MD5는 32 비트 컴퓨터에 최적화되어 있다.

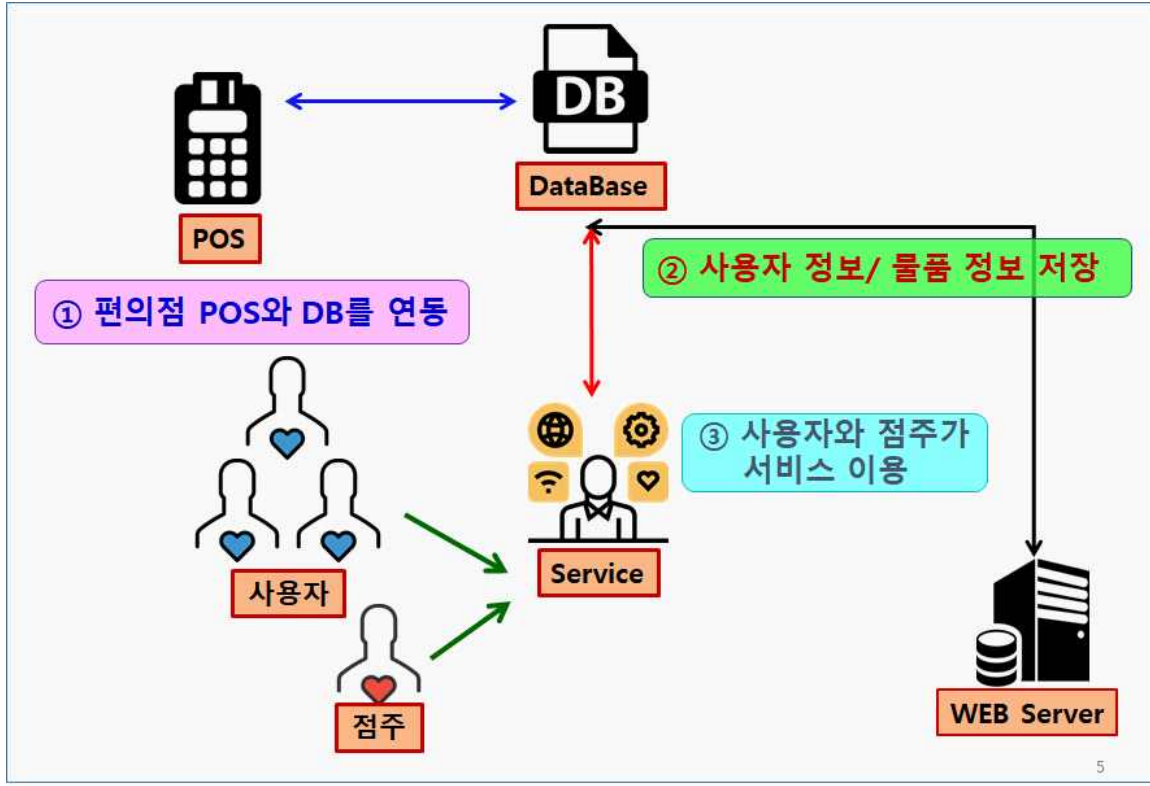
### 2.6 Port forwarding

포트 포워딩(port forwarding) 또는 포트 매핑(port mapping)은 컴퓨터 네트워크에서 패킷이 라우터나 방화벽과 같은 네트워크 게이트웨이를 가로지르는 동안 하나의 IP 주소와 포트 번호 결합의 통신 요청을 다른 곳으로 넘겨주는 네트워크 주소 변환(NAT)의 응용이다. 이 기법은 게이트웨이(외부망)의 반대쪽에 위치한 보호/내부망에 상주하는 호스트에 대한 서비스를 생성하기 위해 흔히 사용되며, 통신하는 목적지 IP 주소와 포트 번호를

내부 호스트에 다시 매핑함으로써 이루어진다.

### 3. 본 론

#### 3.1 기능 설계



#### 3.2 DB 자료구조 설계

##### 3.2.1 전체 DB 테이블

```
mysql> show tables;
+-----+
| Tables_in_cstore |
+-----+
| basket            |
| ceo_info          |
| community_board  |
| customer_info    |
| favorite_product |
| favorite_store   |
| paymentlist      |
| product           |
| purchaselist     |
| store             |
| store_product    |
+-----+
11 rows in set (0.03 sec)
```

### 3.2.2 사용자(고객) 정보 DB

```
mysql> desc customer_info;
```

Field	Type	Null	Key	Default	Extra
idx	int(11)	NO	PRI	NULL	auto_increment
id	varchar(50)	NO		NULL	
password	varchar(50)	NO		NULL	
phone	varchar(50)	NO		NULL	
adult	int(11)	NO		NULL	
mileage	int(11) unsigned	YES		0	
address	varchar(50)	YES		NULL	
name	varchar(50)	NO		NULL	

8 rows in set (0.02 sec)

### 3.2.3 사용자(점주) 정보 DB

```
mysql> desc ceo_info;
```

Field	Type	Null	Key	Default	Extra
idx	int(11)	NO	PRI	NULL	auto_increment
id	varchar(50)	NO		NULL	
password	varchar(50)	NO		NULL	
phone	varchar(50)	NO		NULL	
address	varchar(50)	YES		NULL	
name	varchar(50)	NO		NULL	

6 rows in set (0.02 sec)

### 3.2.4 구매 내역 DB

```
mysql> desc paymentlist;
```

Field	Type	Null	Key	Default	Extra
idx	int(11)	NO	PRI	NULL	auto_increment
product_code	int(11)	NO		NULL	
store_code	int(11)	NO		NULL	
count	int(11)	NO		NULL	
name	varchar(50)	NO		NULL	
price	int(11)	NO		NULL	

6 rows in set (0.02 sec)

### 3.2.5 주문 내역 DB

```
mysql> desc purchaselist;
```

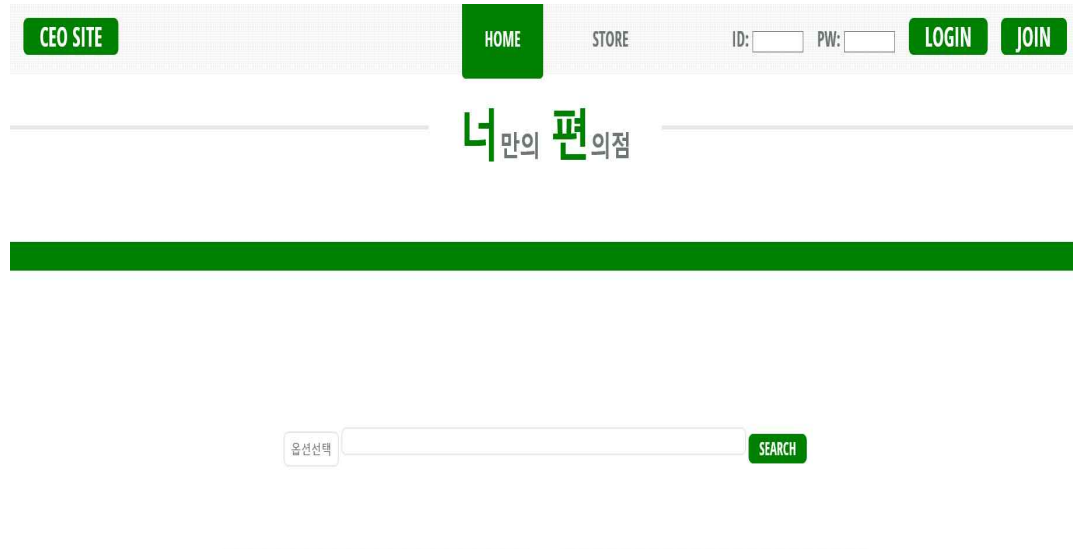
Field	Type	Null	Key	Default	Extra
idx	int(11)	NO	PRI	NULL	auto_increment
product_code	int(11)	NO		NULL	
store_code	int(11)	NO		NULL	
count	int(11)	NO		NULL	
name	varchar(50)	NO		NULL	

5 rows in set (0.00 sec)

### 3.3 구현 상세 내용

#### 3.3.1 초기화면

페이지의 기본 메인 화면이며 로그인, 회원가입, 물품검색 등을 할 수 있다.



#### 3.3.2 새로운 사용자 등록

기본적인 회원가입을 통해 사용자는 페이지의 기본적인 기능들을 사용할 수 있다.



## 회원가입



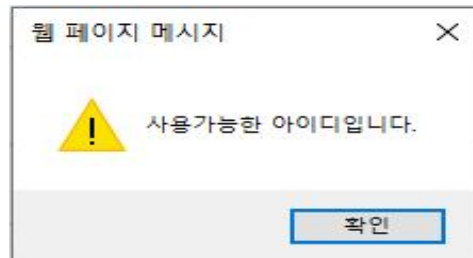


회원가입을 위해 Join을 선택한 후 청소년 또는 성인 회원가입을 선택한다.

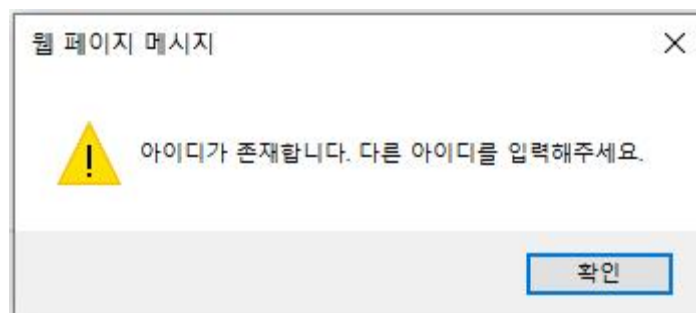
## 회원가입

성명	<input type="text" value="엠태원"/>	
ID	<input type="text" value="sksdudwosla"/>	<input type="button" value="중복확인"/>
비밀번호	<input type="password" value="••••"/>	
비밀번호 확인	<input type="password" value="••••"/>	
핸드폰	<input type="text" value="0102345678"/>	<input type="button" value="X"/>
주소	<input type="text"/>	
상세주소	<input type="text"/>	

회원가입을 위해 필요한 정보를 입력한다.

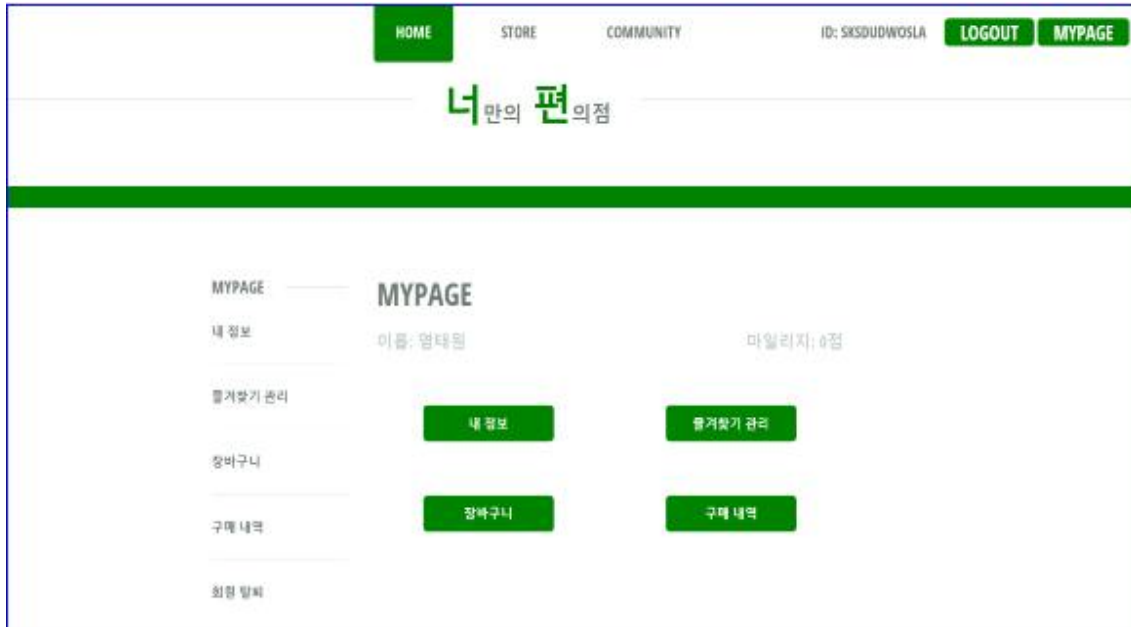


중복확인을 하면 DB의 정보와 비교 후 정보가 존재하지 않으면 위와 같은 팝업창이 뜬다.



중복확인을 하면 DB의 정보와 비교 후 정보가 존재하면 위와 같은 팝업창이 뜬다.

### 3.3.3 마이페이지



마이페이지에서 내정보, 즐겨찾기, 장바구니, 구매내역 등을 확인할 수 있다.

## 내 정보

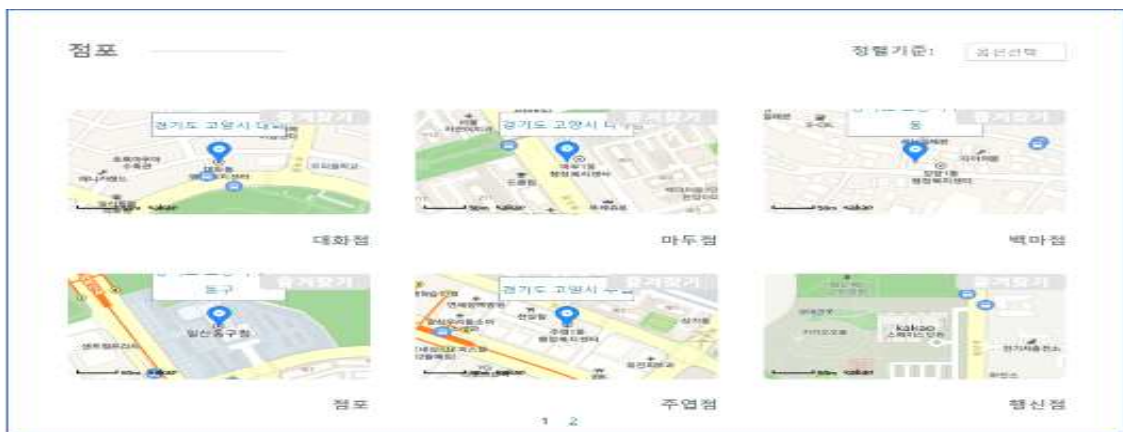
ID	sksdudwosla
이름	영태원
전화번호	01012345678
주소	경기도 파주시 교하읍 가림마을 남양휴븐
마일리지	0

내 정보에서는 상세한 정보를 알 수 있다.

편의점 별

물품 별

즐거찾기관리에서는 사용자가 편의점/물품 정보를 확인할 수 있다.

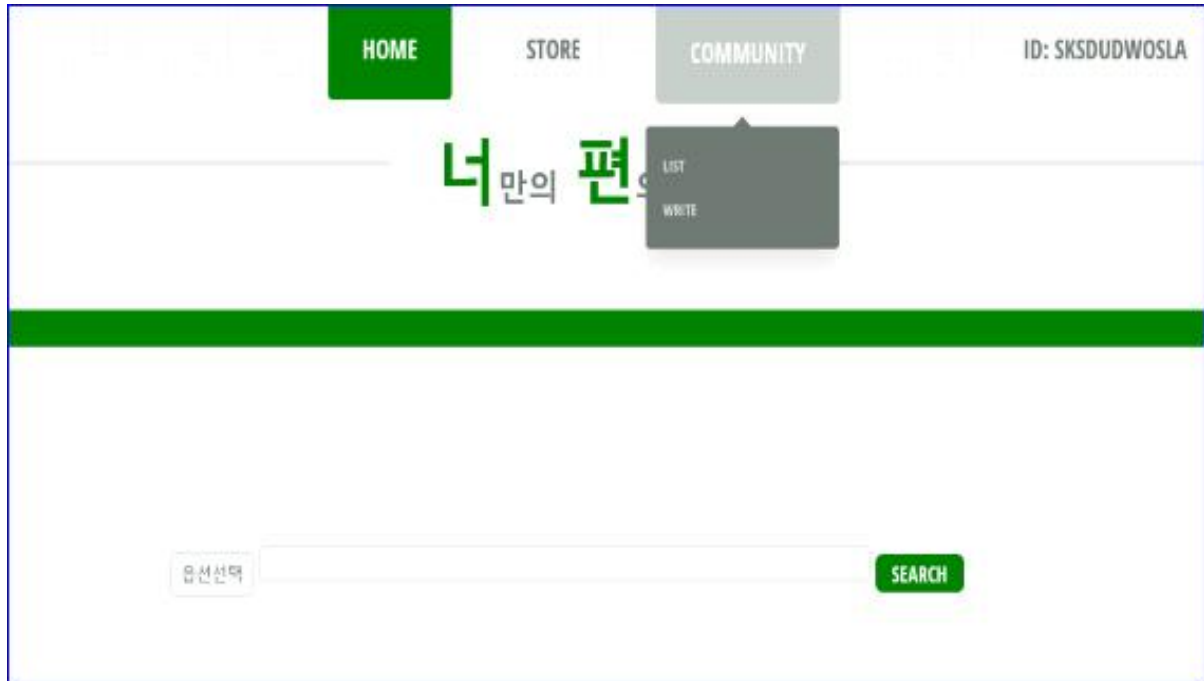


위의 그림은 편의점 별 즐겨찾기이다.



위의 그림은 물품 별 즐겨찾기이다.

### 3.3.4 커뮤니티



커뮤니티에선 사용자들과의 제품 후기나 정보 등을 공유 할 수 있다.



자신만의 후기를 작성해 여러 사람들과 공유할 수 있다.

HOME STORE **COMMUNITY** ID: 0

## 커뮤니티

번호	제목	작성자	작성 날짜
35	안녕하세요	qudrhks	2019-09-06
33	테스트	test	2019-06-25
34	> 테스트	test	2019-06-25
22	한번 더 테스트	test	2019-06-23
20	테스트	test	2019-06-23
19	테스트	test	2019-06-23
18	테스트	test	2019-06-23
17	테스트	test	2019-06-23
32	> 이어서 답글	test2	2019-06-24
30	> 테스트	test2	2019-06-24

1 2 3

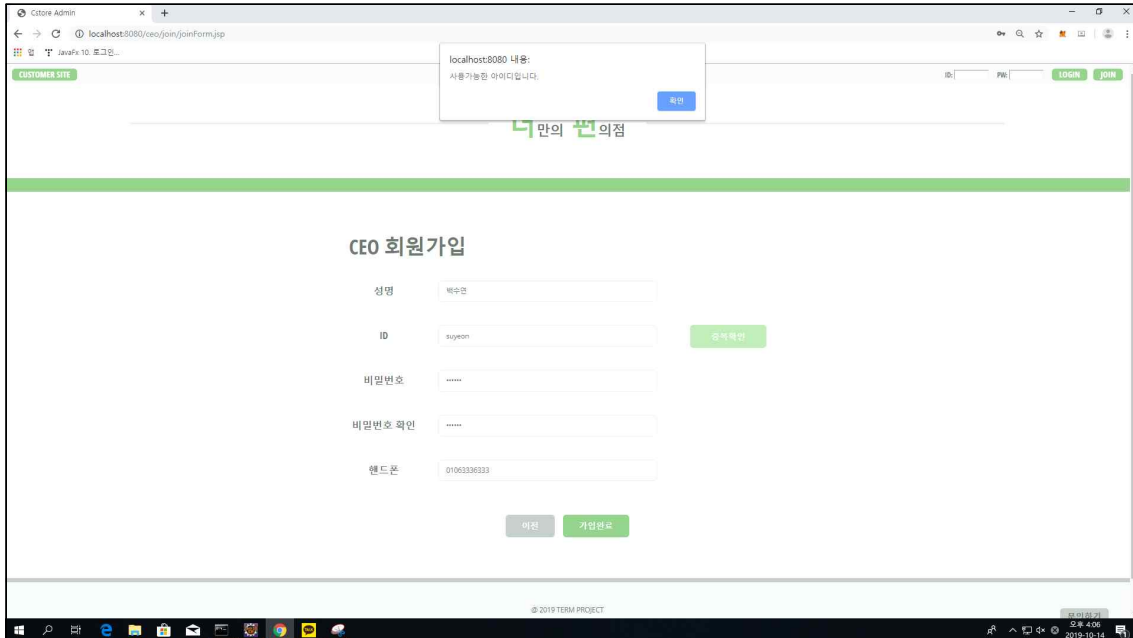
글 작성

위의 그림은 커뮤니티의 글 목록을 보여준다.

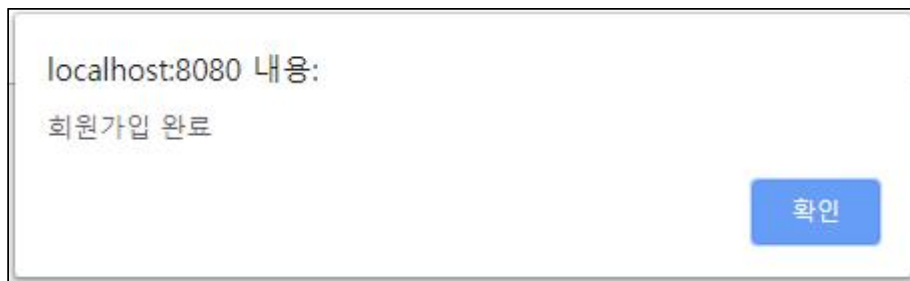
### 3.3.5 새로운 편의점 등록



새로운 편의점을 등록하기 위해서 CEO SITE로 이동한다.



필요한 정보를 입력하고 아이디 중복확인을 통해 CEO 회원가입을 진행한다.



위의 그림과 같은 회원가입 완료 창이 뜬다.



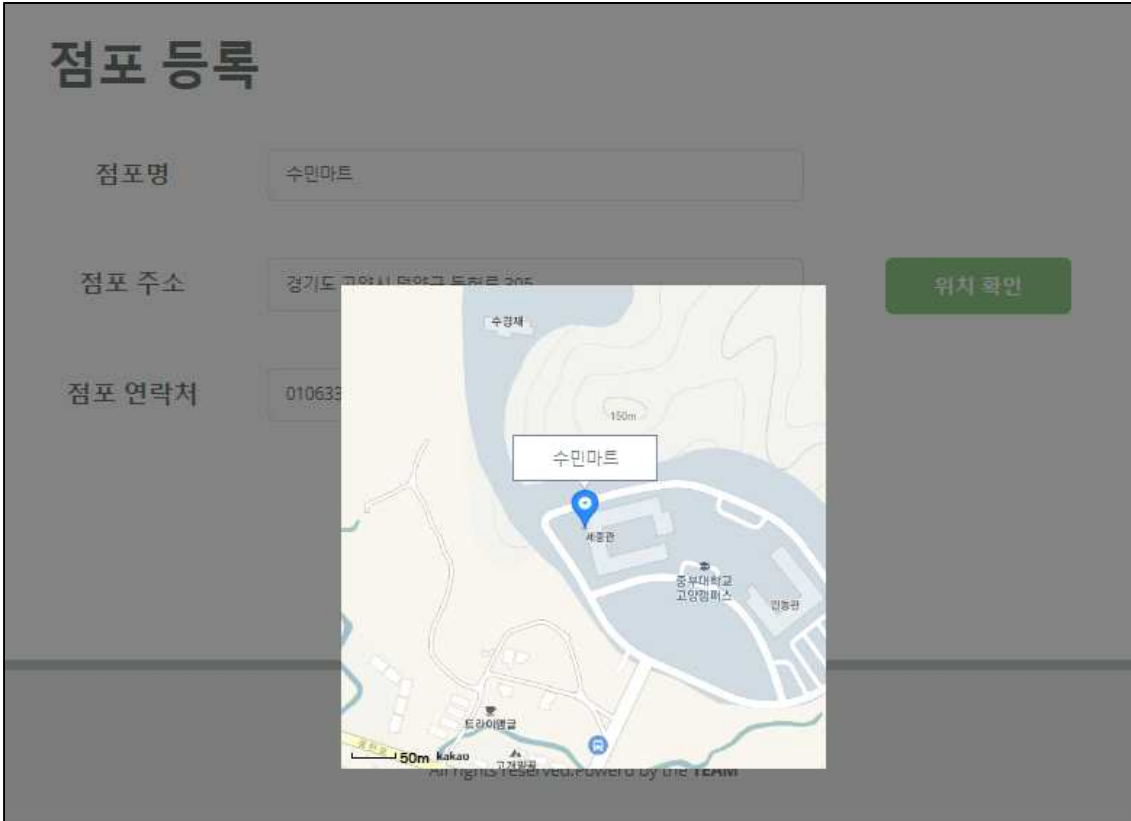
등록한 CEO 계정으로 로그인을 한다.



로그인 후 점포 등록을 선택한다.

A form titled '점포 등록' (Store Registration) in a bold, black, sans-serif font. The form contains three input fields: '점포명' (Store Name) with the value '수민마트', '점포 주소' (Store Address) with the value '경기도 고양시 덕양구 동현로 305', and '점포 연락처' (Store Contact Number) with the value '01063336333'. To the right of the address field is a green button labeled '위치 확인' (Check Location). At the bottom of the form are two buttons: a grey '이전' (Previous) button and a green '등록 완료' (Registration Complete) button.

필요한 정보를 입력한 후 점포 등록을 진행한다.



점포의 주소를 입력한 후 위치를 확인 후 점포 등록을 완료한다.

### 3.3.6 새로운 물품 등록



새로운 물품을 등록하기 위해서 나의 매장을 선택한다.





물품 등록을 선택한다.

### 주문 내역

구매자	제품명	구매 개수	총 가격
skodudwosla	고양근	4	4000원
skodudwosla	블 치즈무도	1	1000원
skodudwosla	원타오렌지	3	3000원
skodudwosla	프랑글루스 오리지날	5	7500원
skodudwosla	어니버터집	3	3000원
borni1004	진라면	3	4500원
borni1004	돼지바	10	5000원
skodudwosla	프리봉	20	20000원
skodudwosla	오징어볶음	8	8000원
borni1004	넵크보이	5	2500원

주문 내역 확인이 가능하다.

### 물품등록

매장 선택

상품명

상품 이미지

상품 가격

매장을 선택한 후 물품을 등록하기 위해 필요한 정보를 입력 후 등록을 완료한다.

## 총합 매출

점포 명	총 판매액
마두점	18500원
대희점	9500원
백마점	14500원
주엽점	25000원
총합 매출	67500원

총합 매출을 확인할 수 있다.

### 3.3.7 마일리지 적립

```
mysql> select * from customer_info;
+----+-----+-----+-----+-----+-----+-----+
| idx | id      | password | phone | adult | mileage | address |
+----+-----+-----+-----+-----+-----+-----+
| 19  | alsrud  | 5d97d8a7d88046a7490e47ceb6d464df | 01055713645 | 1 | 0 | 일산서구 송산로 |
+----+-----+-----+-----+-----+-----+-----+
17 rows in set (0.00 sec)
```

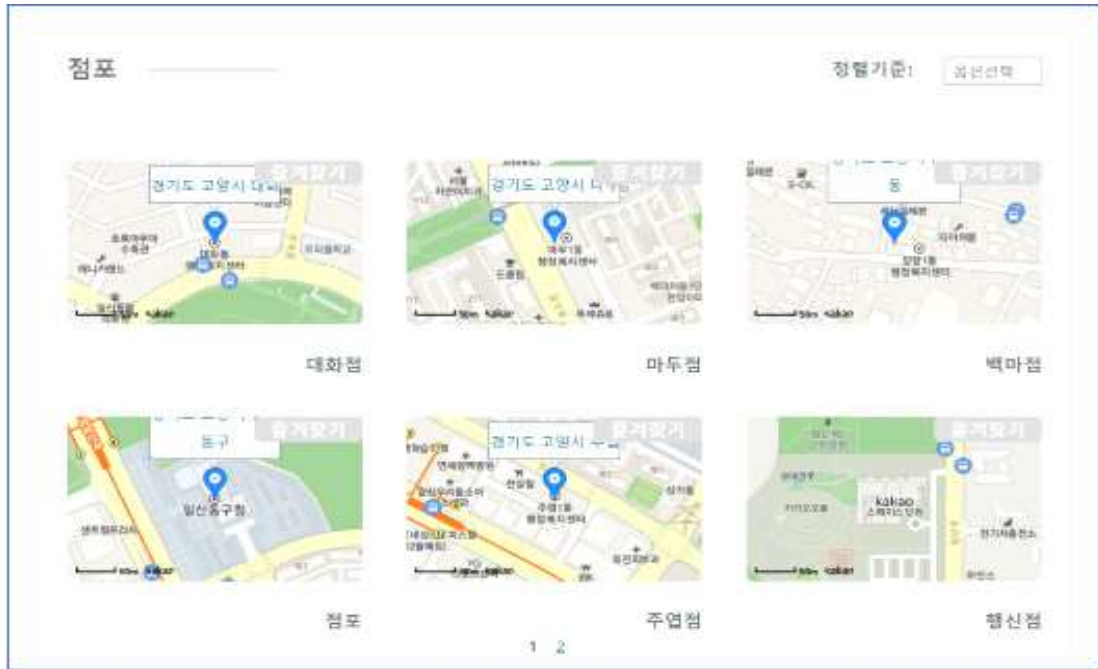
위의 그림은 마일리지를 넣어주기 전의 DB이다.

```
mysql> UPDATE customer_info SET mileage = 200000 WHERE id = 'alsrud';
Query OK, 1 row affected (0.05 sec)
Rows matched: 1 Changed: 1 Warnings: 0

mysql> select * from customer_info;
+----+-----+-----+-----+-----+-----+-----+
| idx | id      | password | phone | adult | mileage | address |
+----+-----+-----+-----+-----+-----+-----+
| 19  | alsrud  | 5d97d8a7d88046a7490e47ceb6d464df | 01055713645 | 1 | 200000 | 일산서구 송산로 |
+----+-----+-----+-----+-----+-----+-----+
17 rows in set (0.00 sec)
```

위의 그림은 마일리지를 넣어준 후의 DB이다.

### 3.3.8 물품구매 및 장바구니



점포를 선택하면 점포에 있는 상품목록을 보여준다.  
원하는 점포를 즐겨찾기 목록에 등록할 수 있다.



점포 선택 후 상품을 선택할 수 있다.



상품 선택 후 수량을 확인하고 구매 및 장바구니에 담을 수 있다.



장바구니를 선택하면 위의 그림과 같은 팝업창이 뜬다.



사용자의 마이페이지의 장바구니에 들어간다.



상품을 장바구니에 담았을 때의 장바구니 화면 결과이다.

### 3.3.9 구매내역 확인

순서	점포명	상품명	가격	수량	QR코드
67	대곡점	진가편	4500원	3개	
68	대곡점	웨이파	5000원	10개	
71	백대점	탈크보미	2500원	5개	
72	백대점	프링어플콘	4000원	4개	
73	주곡점	프리츄	5000원	5개	
74	마두점	포플콘	1000원	1개	

상품을 주문하게 되면 구매 내역에 저장되고 주문했던 내역들을 확인가능하다.

### 3.3.10 비밀번호 해시 값으로 저장

```
/* 핸드폰 입력 정규식 검증 */
var regExp = /(01[016789])([1-9]{1}[0-9]{2,3})([0-9]{4})$/;

if (!regExp.test($("#phone").val())) {
    alert('핸드폰을 입력을 확인하세요. ');
    return false;
}

/* 비밀번호, 비밀번호 확인 검증 */
if($("#join_password").val() != $("#password_Chk").val()) {
    alert('비밀번호 확인이 일치 않습니다. ');
    return false;
}

/* 아이디 중복확인 검증 */
if($("#join_id").val() != $('#input[name=idChecker]').val()) {
    alert('아이디 중복 확인을 해주세요. ');
    return false;
}

$("#join_password").val($.md5($("#join_password").val()));
$("#joinform").submit();
```

비밀번호를 md5 해시 값으로 저장한다.

```
mysql> select * from customer_info;
```

idx	id	password name	phone	adult	mileage	address
1	test	test 테스트	01011111111	0	0000000000	테스트동 테스트로 11
2	test2	test ???	01011111111	1	0000000000	??? ???? 11
4	nonadult	test 청소년테스트	01011111111	0	0000000000	테스트동 테스트로 11
5	adult	test 성인테스트	01011111111	1	0000000000	테스트동 테스트로 11
6	private	2c17c6393771ee3048ae34d6b380c5ec 암호테스트	01011111111	1	0000000000	테스트동 테스트로 11
7	cyj5410	5f658fc3a40153945aa65e68bfa4ec62 최용준	01099486775	1	0000000000	경기도 고양시 일산동구
8	vusdmlwjarh 진달래마을	4b94c46e70e21ecaeee4989ba5bc7741 김병관	01043864100	1	0000000000	경기도 부천시 원미구 상동

7 rows in set (0.02 sec)

사용자 비밀번호가 해시(MD5)로 변환되어 저장된 것을 볼 수 있다.

## 4. 결론 및 향후계획

사용자 입장에서 서비스 시스템을 구현하되, 온라인 주문체계, 개인 정보에 대한 보안 위해 요인을 없애는데 주력했다. 조원 간 끊임없는 토의를 통해 현재 시중에서 사용되고 있는 주문 프로그램들보다 더 좋은 방향으로 나아갈 수 있도록 노력하여 업그레이드된 버전의 프로그램을 개발하였다. 이 서비스를 통해 소비자와 점주 모두에게 편리성 증대 효과를 기대해 볼 수 있다. 전체 조원들이 의기투합하여 이번 졸업 작품 프로젝트를 계획대로 차질 없이 수행하여 중단 없이 운영되는 시스템을 구축하였다.

현재는 시간과 역량의 부족으로 실제 점포의 포스와의 연동은 구현하지 못하였다. 또한 보안적인 요소도 많이 개발하지 못하였다. 향후 실제 점포에서 사용할 수 있도록 개선하고 웹이 가지는 취약점을 보완할 수 있는 쪽으로 개발한다면 더욱 완성도가 높아질 것 같다.

## 5. 첨부

### 5.1 발표 자료

### 5.2 소스코드

## 편의점 온라인 통합 시스템 개발

2019. 10. 29

중부대학교 정보보호학과

지도교수 : 이병천 교수님



3조 최용준  
염태원  
김병관  
박근영  
김민경  
백수연

### 목차



- 조원 편성
- 주제 선정
- 구 상 도
- 추진 경과
- 개발 환경 및 시스템 개발
- 개발 시스템 운영
- 결론 및 기대효과



## 조원 편성



이름	역할
최용준	데이터베이스 구축 및 관리 (총괄)
박근영	프론트엔드 구축 및 개발
염태원	프론트엔드 구축 및 개발, ppt 작성
김병관	프론트엔드 구축 및 개발, ppt 작성
김민경	백엔드 구축 및 개발, 보고서 작성
백수연	백엔드 구축 및 개발, 보고서 작성

3

## 주제 선정



### □ 편의점 온라인 통합 서비스 구현

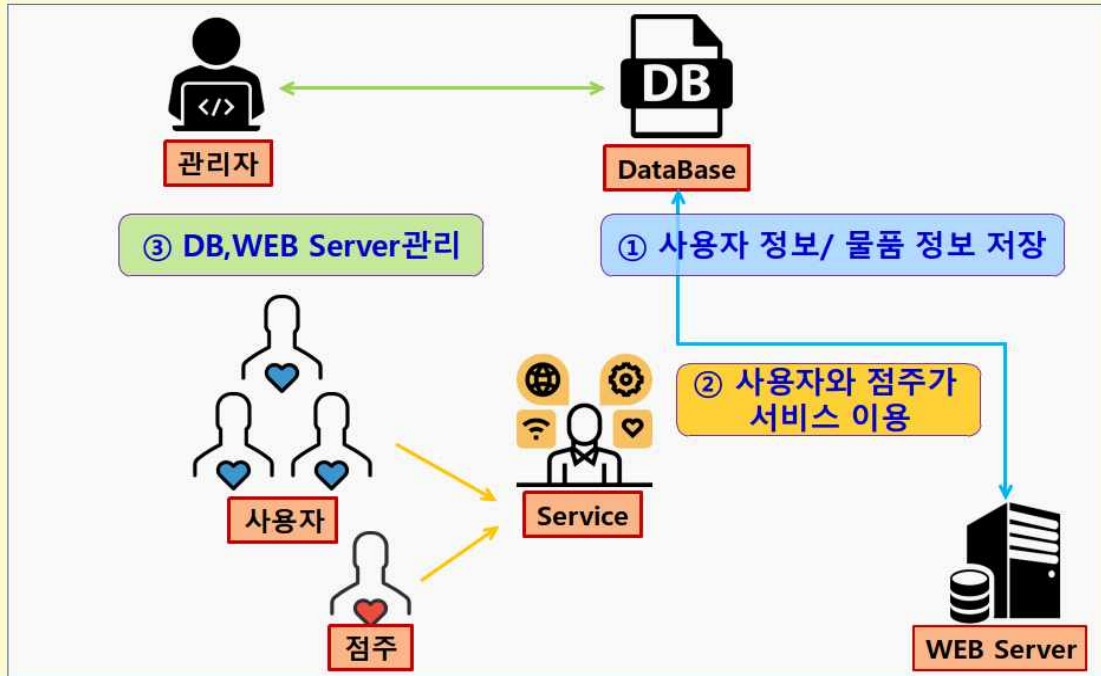
- 시스템을 통해 해당 편의점의 물품 재고를 미리 확인 및 주문 결제 구현
- 직관적인 시스템을 통해 소비자와 점주의 편의성 증대
- 편의점 점주 입장에서는 효율적이고 투명한 편의점 관리가 가능.



점주와 사용자 편의성 극대화를 위해 편리하고 안전한 서비스 체제 구축

4

# 구상도



5

5

# 추진경과



작업	기간 (2019년)								
	3월	4월	5월	6월	7월	8월	9월	10월	
자료 조사	██████████								
DB 구축		████████████████████							
웹 서버 연동		██████████████████							
인증 및 보안 적용			██████████████						
테스트 기간							██████████		
PPT 및 보고서 완성							██████████		

6

# 개발 환경 및 시스템 개발 (1/10)



## 개발 환경

### 운영체제

Windows 10

### 웹 서버

Apache Tomcat → 서버

### 데이터 베이스

MySQL

### 개발 언어

JSP  
JAVA

7

# 개발 환경 및 시스템 개발 (2/10)



## 초기 화면 구현

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html lang="ko">

<jsp:include page="/view/include/head.jsp"/>

<body>
  <div id="page-wrapper">
    <jsp:include page="/view/include/header.jsp"/>
    <jsp:include page="/view/include/nav.jsp" />

    <section id="main" class="main">
      <div id="nav">
        <ul>
          <li class="current"><a href="/">Home</a></li>
          <li><a href="#">Store</a>
            <ul>
              <li><a href="#">List</a></li>
              <li><a href="#">Register</a></li>
              <li><a href="#">Location</a></li>
            </ul>
          </li>
        </ul>
        <div class="login_box left">
          <button class="ceo_btn">CEO SITE</button>
        </div>
        <div class="login_box right">
          <label for="id">ID: </label> <input type="text" id="id" name="id" />
          <label for="pw">PW: </label> <input type="password" id="pw" name="pw" />
          <button class="login_btn">LOGIN</button>
          <button class="join_btn" onclick="location.href='/view/join/joinMain.jsp'">JOIN</button>
        </div>
      </div>
    </section>
  </div>
</body>
</html>
```

초기 화면 설정

상단 고정 바 설정

8

## DB 관리

```

public class DBConnector {
    PreparedStatement pstmt;
    Connection con = null;

    public Connection DBConnector(String url, String user, String pwd) {
        try {
            Class.forName("com.mysql.jdbc.Driver");
            con = DriverManager.getConnection(url, user, pwd);
        } catch (Exception e) {
            System.out.println("DB 접속 실패: " + e.getMessage());
        }
        return con;
    }

    public void ConnectorClose(Connection con, PreparedStatement pstmt) {
        try {
            if (con == null) {
                System.out.println("접속된 Connection이 없습니다.");
            } else {
                con.close();
                System.out.println("접속된 Connection을 종료하였습니다.");
            }

            if (pstmt == null) {
                System.out.println("접속된 PreparedStatement이 없습니다.");
            } else {
                pstmt.close();
                System.out.println("접속된 PreparedStatement을 종료하였습니다.");
            }
        } catch (Exception e) {
            // TODO: handle exception
        }
    }
}
    
```

DB 접속

DB 접속 확인

DB 연결 후 DB 업데이트 등 필요한 작업 실행

9

## 회원 가입 (1/3)

```

<form action="/CustomerController" method="POST" class="form join" id="joinform">
<div class="row">
<div class="col-12 row">
<label for="phone">핸드폰 번호</label>
<input type="text" value="" id="phone"/>
</div>
<div class="col-12 row">
<label for="password">비밀번호</label>
<input type="password" value="" id="password"/>
<input type="password" value="" id="password_chk"/>
</div>
<div class="col-12 row">
<label for="id">아이디</label>
<input type="text" value="" id="id"/>
</div>
</div>
<div class="col-12 row">
<input type="submit" value="회원가입"/>
</div>
</form>

function submitTest() {
    var regexp = /^(01[016789])\{[1-9]\{1\}[0-9]\{2,3\}\}$/;
    if (!regexp.test($("#phone").val())) {
        alert('핸드폰을 입력을 확인하세요. ');
        return false;
    }
    if ($("#join_password").val() != $("#password_chk").val()) {
        alert('비밀번호 확인이 옳지 않습니다. ');
        return false;
    }
    if ($("#join_id").val() != $('#input[name=idChecker]').val()) {
        alert('아이디 중복 확인을 해주세요. ');
        return false;
    }
    $("#joinform").submit();
}
    
```

회원정보 입력

입력 검증

입력 세부 검증

성명, ID 등 회원 정보를 입력받아 검증

10



# 개발 환경 및 시스템 개발 (5/10)



## 회원 가입 (2/3)

```
<div class="col-12 row mgt05">
  <label for="join_id" class="col-3">ID</label>
  <input type="text" id="join_id" name="join_id" class="col-6" />
  <button type="button" onClick="idCheck()" class="button col-2 right">중복확인</button>
  <input type="hidden" name="idChecker" value="unCheck">
</div>
<script>
$(document).ready(function() {
  $('#join').append('<input name="adult" id="adult" style="display:none;" type="checkbox" />');
});

function idCheck() {
  var $id = $('#join_id').val();

  if($id == "") {
    alert('아이디를 입력하세요.');
```

**ID 중복 확인**

**ID 중복 확인함수**

회원정보 입력과정에서 중복 ID를 체크하여 ID를 중복사용한 것으로 확인되면 ID를 변경하도록 안내

11

# 개발 환경 및 시스템 개발 (6/10)



## 회원 가입 (3/3)

```
public class CustomerVO {
  private String name;
  private String id = "";
  private String password = "";
  private String phone = "";
  private String address = "";
}

public class CustomerDB {
  private Connection con;
  private PreparedStatement pstmt;
  private ResultSet rs;
}

public int join(CustomerVO customerVO) {
  String SQL = "INSERT INTO customer_info (id,password,phone,adult,address,name) VALUES ('" + customerVO.getId() + "','" + customerVO.getPassword() + "','" + customerVO.getPhone() + "','" + customerVO.getAdult() + "','" + customerVO.getAddress() + "','" + customerVO.getName() + "')";

  try {
    pstmt = con.prepareStatement(SQL);
    pstmt.setString(1, customerVO.getId());
    pstmt.setString(2, customerVO.getPassword());
    pstmt.setString(3, customerVO.getPhone());
    pstmt.setInt(4, customerVO.getAdult());
    pstmt.setString(5, customerVO.getAddress());
    pstmt.setString(6, customerVO.getName());
    return pstmt.executeUpdate();
  } catch (Exception e) {
    e.printStackTrace();
  }
}
```

**멤버변수 설정**

**DB에 JDBC로 연결**

**DB에 값 추가**

사용자 입력 받은 값을 바탕으로 성명, ID 등 DB 구축에 필요한 회원정보를 입력받아 회원 DB 구축

12

## 비밀번호 해시화

```

/* 비밀번호, 비밀번호 확인 검증 */
if($("#join_password").val() != $("#password").val()) {
    alert('비밀번호 확인이 일치 않습니다. ');
    return false;
}

/* 아이디 중복확인 검증 */
if($("#join_id").val() != $('#input[name=idChecker]').val()) {
    alert('아이디 중복 확인을 해주세요. ');
    return false;
}

$("#join_password").val($.md5($("#join_password").val()));
$("#joinform").submit();
    
```

비밀번호 해시화

암호체계 : MD5(Message Digest 5) 해시함수

암호방식 : 임의 길이의 입력 데이터를 128비트 고정 길이의 해시 값으로 변환

## 암호화 결과

```
mysql> select * from customer_info;
```

idx	id	password name	phone	adult	mileage	address
1	test	test 테스트	01011111111	0	0000000000	테스트동 테스트로 11
2	test2	test ???	01011111111	1	0000000000	???? ???? 11
4	nonadult	test 청소년테스트	01011111111	0	0000000000	테스트동 테스트로 11
5	adult	test 성인테스트	01011111111	1	0000000000	테스트동 테스트로 11
6	private	2c17c6393771ae3048ae34d6b380c5ec 암호테스트	01011111111	1	0000000000	테스트동 테스트로 11
7	cyj5410	5f658fc3e40153945aa65e68bfa4ec62 최용준	01039486775	1	0000000000	경기도 고양시 일산동구
8	vusdm1wjarh 진달래마을	4b94c46e70e21ecaeae4989ba5bc7741 김병관	01043864100	1	0000000000	경기도 부천시 원미구 상동

7 rows in set (0.02 sec)

사용자 비밀번호를 해시(MD5)하여 저장

## 마이페이지

```

<h2>MyPage</h2>
<p class="f_parent">
  <span class="f1">이름: <%=customerVO.getName()%>
  <span class="f2">마일리지: <%=customerVO.getMileage()%>점
</span>
</p>
</header>

<section class="mypage_b">
  <ul class="actions f">
    <li class="f1">정보</li>
    <li class="f2">내역</li>
  </ul>
  <ul class="actions f">
    <li class="f1">정보</li>
    <li class="f2">내역</li>
  </ul>
  <table border="1">
    <tbody>
      <tr>
        <th>ID</th>
        <td><%=customerVO.getId() %></td>
      </tr>
      <tr>
        <th>이름</th>
        <td><%=customerVO.getName() %></td>
      </tr>
      <tr>
        <th>전화번호</th>
        <td><%=customerVO.getPhone()%></td>
      </tr>
      <tr>
        <th>주소</th>
        <td><%=customerVO.getAddress()%></td>
      </tr>
      <tr>
        <th>마일리지</th>
        <td><%=customerVO.getMileage()%></td>
      </tr>
    </tbody>
  </table>
  </div>
  
```

초기 구성 화면

내정보 구성 화면

15

## 커뮤니티

```

<h2>커뮤니티</h2>
</header>
</article>
</div>

<script>
function submitWrite() {
  /* 입력 검증 */
  if ($("#title").val() == "") {
    alert("제목은 입력하세요.");
    return false;
  } else if ($("#contents").val() == "") {
    alert("내용은 입력하세요.");
    return false;
  }

  $(".board").append($("#input name='type' id='type' style='display:none'").val("write"));
  $(".board").append($("#input name='author' id='author' style='display:none'").val("<%=author%"));

  $("#writeform").submit();
}
</script>

<script>
function submitReply() {
  /* 입력 검증 */
  if ($("#reply").val() == "") {
    alert("내용은 입력하세요.");
    return false;
  } else if ($("#contents").val() == "") {
    alert("내용은 입력하세요.");
    return false;
  }

  $(".board").append($("#input name='type' id='type' style='display:none'").val("reply"));
  $(".board").append($("#input name='ref' id='ref' style='display:none'").val("<%=communityVO.getBoardRef() %"));
  $(".board").append($("#input name='level' id='level' style='display:none'").val("<%=communityVO.getBoardLevel() %"));
  $(".board").append($("#input name='step' id='step' style='display:none'").val("<%=communityVO.getBoardStep() %"));
  $(".board").append($("#input name='author' id='author' style='display:none'").val("<%=author%"));

  $("#replyform").submit();
}
</script>
  
```

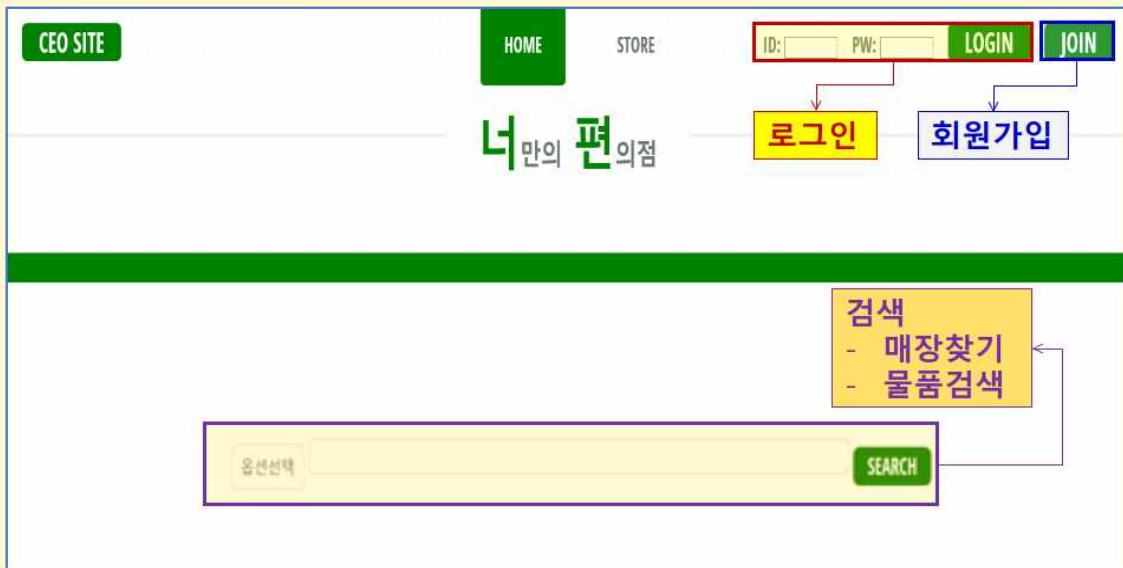
커뮤니티 초기화면

커뮤니티 작성

답변 작성

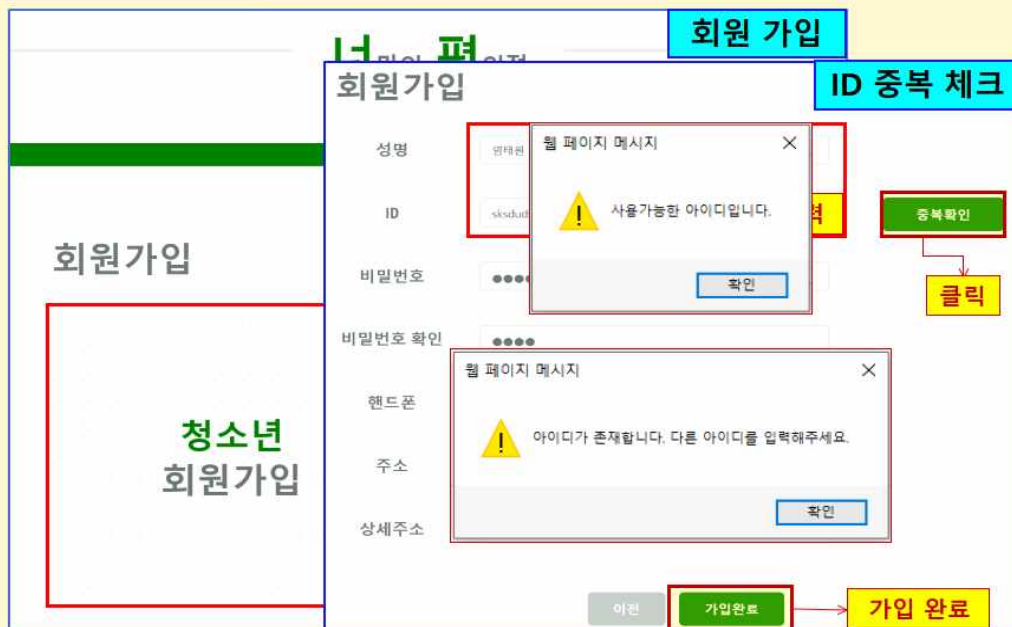
16

## 초기 화면



17

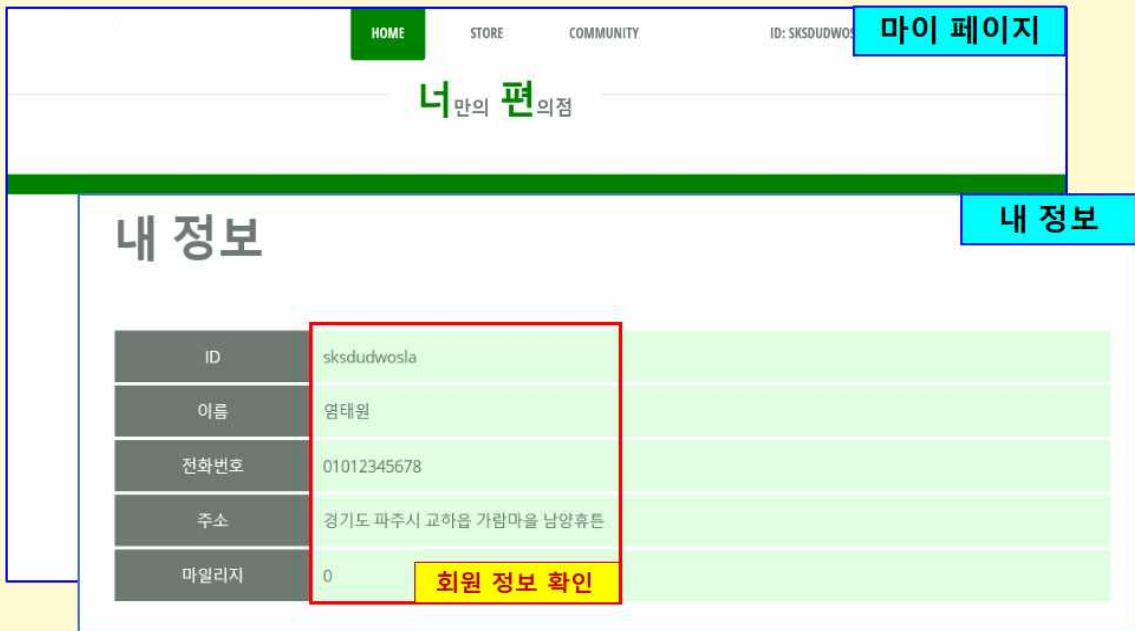
## 회원 가입



18



## 마이페이지

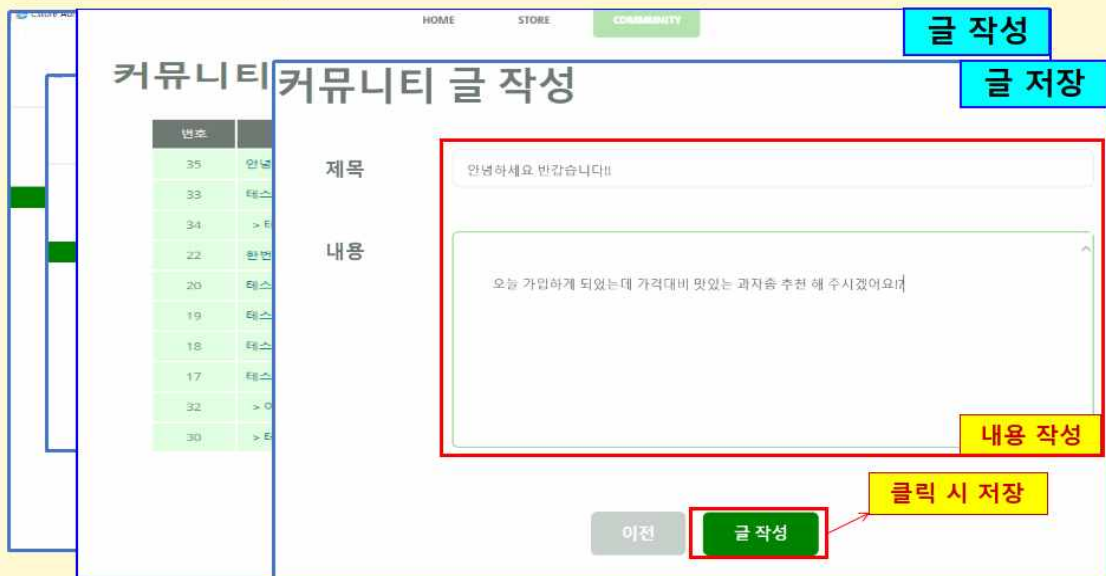


ID	sksdudwosla
이름	염태원
전화번호	01012345678
주소	경기도 파주시 교하읍 가람마을 남양휴튼
마일리지	0

**회원 정보 확인**

19

## 커뮤니티



번호	내용
35	안녕
33	테스
34	> 테
22	환면
20	테스
19	테스
18	테스
17	테스
32	> 테
30	> 테

**제목**

안녕하세요 반갑습니다!!

**내용**

오늘 가입하게 되었는데 가격대비 맛있는 과자좀 추천 해 주시겠어요!

**내용 작성**

**글 작성**

**클릭 시 저장**

20

# 개발 시스템 운영 (5/9)

## 물품구매1

상품목록

정렬기준: 상품목록

상품 상세

포깁콘

가격 1000원

수량 선택 1

총 금액 1000원

구매하기 장바구니 즐겨찾기

장바구니 즐겨찾기

# 개발 시스템 운영 (6/9)

## 물품구매2

구매 내역

구매하기

장바구니

장바구니

즐거찾는 상품

정렬기준: 즐겨찾기

순서	점포명	상품
67	대회점	자일리신
68	대회점	자일리신
71	백마점	자일리신
72	백마점	자일리신
73	주엽점	자일리신
74	마두진	포깁콘

즐거찾기 해제

즐거찾기 해제

즐거찾기 해제

즐거찾기 추가

포깁콘

코카콜라

멜치스포도

즐거찾기 해제

프링글스 오리지날

## 매장관리 1

너만의 편의점
매장관리

점포 등록

점포등록

점포명


점포 주소

점포 연락처

마두역

경기도 파주시 미래로 564

01012345678



마두역

054역대명

위치 확인

위치확인

이전

등록 완료

매장등록

23

## 매장관리 2

너만의 편의점
매장관리

너만의 편의점

나의매장

주문  
내역

총합  
매출

물품  
등록

클릭

24

## 매장관리 3

주문 내역		주문내역
물품등록		물품등록
총합 매출		총매출
점포 명	총 판매액	
마두점	18500원	
대화점	9500원	
백마점	14500원	
주엽점	25000원	
총합 매출	67500원	매장 별 매출 내역

25

# 결론 및 기대효과

### □ 결 론

- 회원관리용 DB 구축 및 연동, 가상결제 등에 중점을 둔 편의점 온라인 통합시스템 개발에 성공
- 특히 개발 기간중의 일부 조원 교육파견 등 인력 운영의 어려움을 극복하고 편의점 웹 페이지를 차질없이 구현

### □ 기대 효과

- 모든 조원이 임무를 분담하여 웹 구축, DB 구축 및 연동, 보안 기능 등을 자체 구현함으로써 시스템 기획 및 프로그램 개발 역량을 배양
- 개발 시스템은 소비자와 점주 모두에게 편리성을 증대시키는 효과가 기대. 끝.

26

---

**Q & A**

**감사합니다.**

---

## 5.2 프로그램 소스코드 일부

### 5.2.1 초기화면

```
index.jsp

<%@page import="java.util.List"%>
<%@page import="java.util.ArrayList"%>
<%@page import="com.pjt.cstore.StoreDAO"%>
<%@page import="com.pjt.cstore.StoreVO"%>
<%@page import="com.pjt.cstore.ProductDAO"%>
<%@page import="com.pjt.cstore.ProductVO"%>
<%@page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html lang="ko">

<%
    String ceo = (String) session.getAttribute("ceo");
    String user = (String) session.getAttribute("user");

    if (ceo != null && ceo.equals("1")) {
        response.sendRedirect("/ceo");
    }

    StoreDAO storeDAO = new StoreDAO();
    StoreVO store = null;
    store = storeDAO.favoritestore(user);

    ProductDAO productDAO = new ProductDAO();
    ProductVO product,newProduct,saleProduct = null;

    product = productDAO.favoriteproduct(user);
    newProduct = productDAO.newproduct();
    saleProduct = productDAO.saleproduct();
%>

<jsp:include page="/include/head.jsp" />

<body class="homepage is-preload">
    <div id="page-wrapper">
        <jsp:include page="/include/header.jsp" />
        <jsp:include page="/include/nav.jsp" />
        <section id="main" class="search">
            <div class="container">
                <div class="row gtr-200">
                    <div class="col-12">
                        <section class="box highlight search_box">
                            <form action="/store/search.jsp"
method="GET" class="SearchForm">
```

```

id="searchType" >
value="store">옵션선택</option>
value="store">매장찾기</option>
value="product">물품검색</option>
type="text" class="search" name="search" title="검색" />
class="search_btn">SEARCH</button>
</form>
</section>
</div>
<div class="col-12 itemThumbnail">
<section class="box features">
<div>
<div class="row">
<div class="col-6
col-6-medium col-12-small">
class="box feature">
<% if(newProduct != null) { %>
href="/store/newproduct.jsp" class="image featured main_favorite"> </a>
<% } else { %>
href="/store/newproduct.jsp" class="image featured"> </a>
<% } %>
<h3>
<a href="/store/newproduct.jsp">+ 신규 상품 더보기</a>
</h3>
</section>
</div>
<div class="col-6
col-6-medium col-12-small">
class="box feature">
<% if(saleProduct != null) { %>
href="/store/newproduct.jsp" class="image featured main_favorite"> </a>
<% } else { %>
href="/store/newproduct.jsp" class="image featured"> <a
        src="/assets/img/pic01.jpg" alt="" /> </a>
<% } %>
<h3>
    <a href="/store/saleproduct.jsp">+ 할인 상품 더보기</a>
</h3>
</section>
</div>
</div>
</div>
</section>
</div>
<%
    if (user != null) {
%>
<div class="col-12">
    <section class="box features">
        <h2 class="major">
            <span>즐거찾기</span>
        </h2>
        <div>
            <div class="row">
                <div class="col-6
col-6-medium col-12-small">
                    <%
                        if
(store.getFavorite() == 0) {
                            <section class="box feature">
                                <a
href="#" class="image featured main_favorite"> <a
                                    src="/assets/img/pic01.jpg" alt="" /> </a>
                                <h3>
                                    <a href="#">편의점</a>
                                </h3>
                            </section>
                        <%
                            }
                    %>
                else {

```



<script>

```
function locationSet(id, address, name) {
    $('.'+id).css('display', 'block');
    var mapContainer = document.getElementById(id),
        mapOption = {
            center : new kakao.maps.LatLng(33.450701, 126.570667),
            level : 3
        };

    var map = new kakao.maps.Map(mapContainer, mapOption);

var geocoder = new kakao.maps.services.Geocoder();
    geocoder.addressSearch(
        address,
        function(result, status) {
            if
(status === kakao.maps.services.Status.OK) {
                var coords = new kakao.maps.LatLng(
                    result[0].y, result[0].x);

var marker = new kakao.maps.Marker({
                    position : coords
                });

var infowindow = new kakao.maps.InfoWindow(
                    {
                        content : '<div
style="width:150px;text-align:center;padding:6px 0;">'+address+'</div>'
                    }
                );
            }
        }
    );
}
```

```

        });
        infowindow.open(map, marker);
        map.setCenter(coords);
        $('input[name=locChecker]').val(" check");
    }

});

    }
</script>
<section
class="box feature">
    <a
href="/store/product.jsp?code=<%=store.getCode()%>"
class="image featured main_favorite"
id="map<%=store.getCode()%>"></a>
<h3>
    <a href="/store/product.jsp?code=<%=store.getCode()%>"><%=store.getName()%></a>
</h3>
</section>
<script>
    locationSet("map<%=store.getCode()%>", "<%=store.getAddress()%>",
"<%=store.getName()%>");
</script>
<%
    }
%>
</div>
<div class="col-6
col-6-medium col-12-small">
    <%
    if
(product.getFavorite() == 0) {
    %>
<section
class="box feature">
    <a
href="#" class="image featured"><img

```



```
</html>
```

## 5.2.2 새로운 사용자 등록

### joinMain.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html lang="ko">

<jsp:include page="/include/head.jsp"/>

<body>
    <div id="page-wrapper">

        <jsp:include page="/include/header.jsp" />

        <jsp:include page="/include/nav.jsp" />

        <section id="main">
            <div class="container">
                <div class="col-12">

                    <div class="content">

                        <article class="box page-content">

                            <header>
                                <h2>회원가입</h2>
                            </header>

                            <section class="box features">
                                <div>
                                    <div class="row">
                                        <div
class="col-6">
<section class="box feature">
    <h2>
        <a href="/joinForm.jsp?adult=0" class="image featured joinBtn">
            <span class="tGreen">청소년</span> <br>회원가입
        </a>
    </h2>
</section>
                                        </div>
                                        <div
class="col-6">
<section class="box
```



```

        adult = "0";
    }
%>
<div id="page-wrapper">

    <jsp:include page="/include/header.jsp" />

    <jsp:include page="/include/nav.jsp" />

    <section id="main">
        <div class="container">
            <div class="col-12">

                <div class="content">

                    <article class="box page-content">

                        <header>
                            <h2>회원가입</h2>
                        </header>

                        <section class="box features">
                            <div>
                                <form
action="/CustomerJoin" method="POST"
                                class="form
join" id="joinform">
                                <div
class="row">
<div class="col-12 row mgt05">
                <label for="name" class="col-3">성명</label> <input
                type="text" id="name" name="name" class="col-6" required />
</div>
<div class="col-12 row mgt05">
                <label for="join_id" class="col-3">ID</label> <input
                type="text" id="join_id" name="join_id" class="col-6"
                required />
                <button type="button" onClick="idCheck()"
                class="button col-2 right">중복확인</button>
                <input type="hidden" name="idChecker" value="unCheck">
</div>
<div class="col-12 row mgt05">
                <label for="join_password" class="col-3">비밀번호</label> <input

```

```

        type="password" id="join_password" name="join_password"
        class="col-6" required />
</div>
<div class="col-12 row mgt05">
    <label for="password_Chk" class="col-3">비밀번호 확인</label> <input
        type="password" id="password_Chk" name="password_Chk"
        class="col-6" required />
</div>
<div class="col-12 row mgt05">
    <label for="phone" class="col-3">핸드폰</label> <input
        type="text" id="phone" name="phone" class="col-6" maxlength="11" required
    />
</div>
<div class="col-12 row mgt05">
    <label for="address" class="col-3">주소</label> <input
        type="text" id="address" name="address" class="col-6"
        required />
</div>
<div class="col-12 row mgt05">
    <label for="detail_address" class="col-3">상세주소</label> <input
        type="text" id="detail_address" name="detail_address"
        class="col-6" required />
</div>
<div class="col-12">
    <ul class="actions">
        <li><a href="javascript:history.back()"
            class="button alt">이전</a> </li>
        <li><button type="button" onClick="submitJoin()"

```





```

        if ($("#name").val() == "") {
            alert('이름을 입력하세요.');
```

```

            return false;
        } else if ($("#join_id").val() == "") {
            alert('아이디를 입력하세요.');
```

```

            return false;
        } else if ($("#join_password").val() == "") {
            alert('비밀번호를 입력하세요.');
```

```

            return false;
        } else if ($("#password_Chk").val() == "") {
            alert('비밀번호 확인을 입력하세요.');
```

```

            return false;
        } else if ($("#phone").val() == "") {
            alert('핸드폰을 입력하세요.');
```

```

            return false;
        } else if ($("#address").val() == "") {
            alert('주소를 입력하세요.');
```

```

            return false;
        } else if ($("#detail_address").val() == "") {
            alert('상세주소를 입력하세요.');
```

```

            return false;
        }

        var regExp = /(01[016789])([1-9]{1}[0-9]{2,3})([0-9]{4})$/;

        if (!regExp.test($("#phone").val())) {
            alert('핸드폰을 입력을 확인하세요.');
```

```

            return false;
        }

        if ($("#join_password").val() != $("#password_Chk").val()) {
            alert('비밀번호 확인이 옳지 않습니다.');
```

```

            return false;
        }

        if ($("#join_id").val() != $('input[name=idChecker]').val()) {
            alert('아이디 중복 확인을 해주세요.');
```

```

            return false;
        }

        $("#join_password").val($.md5($("#join_password").val()));

        $("#joinform").submit();
    }
</script>
</div>
</body>
</html>

```

### CustomerJoin.java

```

package com.pjt.cstore;

import java.io.IOException;
import java.io.PrintWriter;

import javax.servlet.ServletException;

```

```

import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

@WebServlet("/CustomerJoin")
public class CustomerJoin extends HttpServlet {
    private static final long serialVersionUID = 1L;

    protected void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        // TODO Auto-generated method stub
        response.getWriter().append("Served at: ").append(request.getContextPath());
    }

    protected void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        // TODO Auto-generated method stub
        response.setContentType("text/html;charset=utf-8");
        request.setCharacterEncoding("utf-8");

        String name = "";
        String id = "";
        String password = "";
        String phone = "";
        String address = "";
        int adult = 0;
        int result = 0;

        name = request.getParameter("name");
        id = request.getParameter("join_id");
        password = request.getParameter("join_password");
        phone = request.getParameter("phone");
        address = request.getParameter("address") + " " +
request.getParameter("detail_address");
        adult = Integer.parseInt(request.getParameter("adult"));

        if(!(name.equals("") || id.equals("") || password.equals("") || phone.equals("") ||
address.equals(""))) {
            CustomerDAO customerDAO = new CustomerDAO();
            CustomerVO customerVO = new CustomerVO();

            customerVO.setName(name);
            customerVO.setId(id);
            customerVO.setPassword(password);
            customerVO.setPhone(phone);
            customerVO.setAddress(address);
            customerVO.setAdult(adult);

            result = customerDAO.join(customerVO);

            if(result > 0) {
                PrintWriter out = response.getWriter();
                out.println("<script type='text/javascript'>");
                out.println("alert('회원가입 완료');");
                out.println("location.href='/');");
            }
        }
    }
}

```



```

        return phone;
    }
    public void setPhone(String phone) {
        this.phone = phone;
    }
    public String getAddress() {
        return address;
    }
    public void setAddress(String address) {
        this.address = address;
    }
    public int getAdult() {
        return adult;
    }
    public void setAdult(int adult) {
        this.adult = adult;
    }
    public String getOrderProduct() {
        return orderProduct;
    }
    public void setOrderProduct(String orderProduct) {
        this.orderProduct = orderProduct;
    }
    public int getOrderCount() {
        return orderCount;
    }
    public void setOrderCount(int orderCount) {
        this.orderCount = orderCount;
    }
    public int getOrderPrice() {
        return orderPrice;
    }
    public void setOrderPrice(int orderPrice) {
        this.orderPrice = orderPrice;
    }
}

```

### **CustomerDAO.java**

```

package com.pjt.cstore;

import java.math.BigInteger;
import java.security.MessageDigest;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;

import com.pjt.cstore.common.DBConnector;

public class CustomerDAO {

    private Connection con;

    private PreparedStatement pstmt;

```

```

private ResultSet rs;

public CustomerDAO() {
    try {
        DBConnector connector = new DBConnector();
        con = connector.DBCon();
    } catch (Exception e) {
        e.printStackTrace();
    }
}

public int login(String id, MessageDigest password) {
    String SQL = "SELECT password FROM customer_info WHERE id = ?";

    try {
        pstmt = con.prepareStatement(SQL);
        pstmt.setString(1, id);
        rs = pstmt.executeQuery();

        if (rs.next()) {
            if (rs.getString(1).equals(new BigInteger(1,
password.digest()).toString(16))) {
                return 1;
            } else {
                return 0;
            }
        }

        return -1;
    } catch (Exception e) {
        e.printStackTrace();
    }

    return -2;
}

public int join(CustomerVO customerVO) {
    String SQL = "INSERT INTO customer_info
(id,password,phone,adult,address,name) VALUES (?,?,,?,?,,?)";

    try {
        pstmt = con.prepareStatement(SQL);
        pstmt.setString(1, customerVO.getId());
        pstmt.setString(2, customerVO.getPassword());
    }
}

```

```

        pstmt.setString(3, customerVO.getPhone());
        pstmt.setInt(4, customerVO.getAdult());
        pstmt.setString(5, customerVO.getAddress());
        pstmt.setString(6, customerVO.getName());
        return pstmt.executeUpdate();
    } catch (Exception e) {
        e.printStackTrace();
    }
    return -1;
}

public CustomerVO searchAll(String id) {
    CustomerVO customerVO = new CustomerVO();
    String SQL = "SELECT * FROM customer_info WHERE id = ?";

    try {
        pstmt = con.prepareStatement(SQL);
        pstmt.setString(1, id);
        rs = pstmt.executeQuery();

        if (rs.next()) {
            customerVO.setAddress(rs.getString("address"));
            customerVO.setAdult(rs.getInt("adult"));
            customerVO.setId(rs.getString("id"));
            customerVO.setMileage(rs.getInt("mileage"));
            customerVO.setName(rs.getString("name"));
            customerVO.setPhone(rs.getString("phone"));
        }
    } catch (Exception e) {
        e.printStackTrace();
    }

    return customerVO;
}

public int delete(String id) {
    String SQL = "DELETE FROM customer_info WHERE id = (?)";

    try {
        pstmt = con.prepareStatement(SQL);
        pstmt.setString(1, id);
        pstmt.executeUpdate();
        return 1;
    } catch (Exception e) {
        e.printStackTrace();
    }
}

```

```

        }
        return -1;
    }
}

```

### 5.2.3 새로운 편의점 등록

```

updateStoreForm.jsp
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html lang="ko">

<%
    String ceo = (String) session.getAttribute("ceo");
    String id = (String) session.getAttribute("user");

    if (ceo == null) {
        response.sendRedirect("/");
    }
%>

<jsp:include page="/include/head.jsp" />

<body>
    <div class="dim">
        <div id="storeMap"> </div>
    </div>
    <div id="page-wrapper">

        <jsp:include page="/include/header.jsp" />

        <jsp:include page="/include/nav.jsp" />

        <section id="main">
            <div class="container">
                <div class="col-12">

                    <div class="content">

                        <article class="box page-content">

                            <header>
                                <h2>점포 등록</h2>
                            </header>

                            <section class="box features">
                                <div>
                                    <form
action="/StoreJoin" method="POST" class="form join"
id="joinform">
                                                                    <div
class="row">

```

```

<div class="col-12 row mgt05">
    <label for="storeName" class="col-3">점포명</label> <input
        type="text" id="storeName" name="storeName" class="col-6" required />
</div>
<div class="col-12 row mgt05">
    <label for="storeAddress" class="col-3">점포 주소</label> <input
        type="text" id="storeAddress" name="storeAddress" class="col-6"
placeholder="ex: 제주특별자치도 제주시 첨단로 242"
        required />
    <button type="button" onClick="locationCheck()"
        class="button col-2 right">위치 확인</button>
    <input type="hidden" name="locChecker" value="unCheck">
</div>
<div class="col-12 row mgt05">
    <label for="storeCall" class="col-3">점포 연락처</label> <input
        type="text" id="storeCall" name="storeCall"
        class="col-6" required />
</div>
<div class="col-12">
    <ul class="actions">
        <li><a href="javascript:history.back()"
            class="button alt">이전</a></li>
        <li><button type="button" onClick="submitJoin()"
            class="button">등록 완료</button></li>
    </ul>
</div>
</div>
</div>
</form>
</div>
</section>

```



```

</article>
</div>
</div>
</div>
</section>
<jsp:include page="/include/footer.jsp" />
<script>
    $('dim').click(function(e) {
        $('dim').css('display', 'none');
    });

    function locationCheck() {
        $('dim').css('display', 'block');
        var mapContainer = document.getElementById('storeMap'),
            mapOption = {
                center : new kakao.maps.LatLng(33.450701,
126.570667),
                level : 3
            };

        var map = new kakao.maps.Map(mapContainer, mapOption);
        var geocoder = new kakao.maps.services.Geocoder();

        geocoder
            .addressSearch(
                $("#storeAddress").val(),
                function(result, status) {
                    if (status ===
kakao.maps.services.Status.OK) {
                        var coords
= new kakao.maps.LatLng(
                            result[0].y, result[0].x);
                        var marker
= new kakao.maps.Marker({
                            map : map,
                            position : coords
                        });
                        var
infowindow = new kakao.maps.InfoWindow(
                            {
                                content : '<div style="width:150px;text-align:center;padding:6px
0;">'+$('input[name=storeName']).val()+'</div>'
                            });

```

```

infowindow.open(map, marker);

map.setCenter(coords);

$('input[name=locChecker]').val("check");
}

});

}

function submitJoin() {

    if ($("#storeName").val() == "") {
        alert('점포명을 입력하세요.');
```

```

        return false;
    } else if ($("#storeAddress").val() == "") {
        alert('점포 주소를 입력하세요.');
```

```

        return false;
    } else if ($("#storeCall").val() == "") {
        alert('점포 연락처를 입력하세요.');
```

```

        return false;
    }

    if ($('#input[name=locChecker]').val() != "check") {
        alert('점포 위치 확인을 해주세요.');
```

```

        return false;
    }

    $('#join').append($('#<input name="owner" id="owner"
style="display:none"/>').val('<%=id%>'));

    $('#joinform').submit();
}
</script>
</div>
</body>
</html>

```

### StoreJoin.java

```

package com.pjt.cstore;

import java.io.IOException;
import java.io.PrintWriter;

import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

@WebServlet("/StoreJoin")
public class StoreJoin extends HttpServlet {
    private static final long serialVersionUID = 1L;

```

```

public StoreJoin() {
    super();
    // TODO Auto-generated constructor stub
}
protected void doGet(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    // TODO Auto-generated method stub
    response.getWriter().append("Served at: ").append(request.getContextPath());
}

protected void doPost(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    // TODO Auto-generated method stub
    response.setContentType("text/html;charset=utf-8");
    request.setCharacterEncoding("utf-8");

    String storeName = "";
    String storeAddress = "";
    String storeCall = "";
    String owner = "";
    int result = 0;

    storeName = request.getParameter("storeName");
    storeAddress = request.getParameter("storeAddress");
    storeCall = request.getParameter("storeCall");
    owner = request.getParameter("owner");

    if (!(storeName.equals("") || storeAddress.equals("") || storeCall.equals(""))) {
        StoreDAO storeDAO = new StoreDAO();
        StoreVO storeVO = new StoreVO();

        storeVO.setOwner(owner);
        storeVO.setAddress(storeAddress);
        storeVO.setCall(storeCall);
        storeVO.setName(storeName);

        result = storeDAO.join(storeVO);

        if (result > 0) {
            PrintWriter out = response.getWriter();
            out.println("<script type='text/javascript'>");
            out.println("alert('점포 등록 완료');");
            out.println("location.href='/ceo'");
            out.println("</script>");
        } else {
            System.out.println("DB 입력 실패");
        }
    } else {
        PrintWriter out = response.getWriter();
        out.println("<script type='text/javascript'>");
        out.println("alert('입력값에 오류가 있습니다.');");
        out.println("location.href='/';");
        out.println("</script>");
    }
}
}

```

## StoreDAO.java

```
package com.pjt.cstore;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.util.ArrayList;

import com.pjt.cstore.common.DBConnector;

public class StoreDAO {

    private Connection con, con2;

    private PreparedStatement pstmt, pstmt2;

    private ResultSet rs, rs2;

    public StoreDAO() {

        try {

            DBConnector connector = new DBConnector();
            con = connector.DBCon();
            con2 = connector.DBCon();

        } catch (Exception e) {

            e.printStackTrace();

        }

    }

    public int join(StoreVO storeVO) {
        String SQL = "";
        int ref = 0;

        SQL = "SELECT AUTO_INCREMENT FROM information_schema.tables
WHERE TABLE_NAME = 'store' AND table_schema = DATABASE( )";
        try {

            pstmt = con.prepareStatement(SQL);
            rs = pstmt.executeQuery();

            if (rs.next()) {
                ref = rs.getInt(1);
            }
        } catch (Exception e) {
            e.printStackTrace();
        }

        SQL = "INSERT INTO store
(store_owner,store_name,store_address,store_call,code) VALUES (?,?,?,?,?)";

        try {
```

```

        pstmt = con.prepareStatement(SQL);
        pstmt.setString(1, storeVO.getOwner());
        pstmt.setString(2, storeVO.getName());
        pstmt.setString(3, storeVO.getAddress());
        pstmt.setString(4, storeVO.getCall());
        pstmt.setInt(5, ref * 100);
        return pstmt.executeUpdate();

    } catch (Exception e) {

        e.printStackTrace();

    }

    return -1;
}

public ArrayList<StoreVO> list() {
    String SQL = "";
    ArrayList<StoreVO> storeList = new ArrayList<StoreVO>();

    SQL = "SELECT * FROM store ORDER BY store_name ASC";
    try {

        pstmt = con.prepareStatement(SQL);
        rs = pstmt.executeQuery();

        while (rs.next()) {
            StoreVO storeVO = new StoreVO();
            storeVO.setName(rs.getString("store_name"));
            storeVO.setAddress(rs.getString("store_address"));
            storeVO.setCode(rs.getInt("code"));
            storeVO.setFavorite(0);

            storeList.add(storeVO);
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
    return storeList;
}

public ArrayList<StoreVO> customerlist(String user) {
    String SQL = "", SQL2 = "";
    ArrayList<StoreVO> storeList = new ArrayList<StoreVO>();

    SQL = "SELECT * FROM store ORDER BY store_name ASC";
    try {

        pstmt = con.prepareStatement(SQL);
        rs = pstmt.executeQuery();

        while (rs.next()) {
            StoreVO storeVO = new StoreVO();
            storeVO.setName(rs.getString("store_name"));
            storeVO.setAddress(rs.getString("store_address"));

```

```

        storeVO.setCode(rs.getInt("code"));

        SQL2 = "SELECT EXISTS (SELECT * FROM favorite_store
WHERE name = '" + user + "' AND store_code = "
                + rs.getInt("code") + ") AS success";
        pstmt2 = con2.prepareStatement(SQL2);
        rs2 = pstmt2.executeQuery();

        while (rs2.next()) {
            storeVO.setFavorite(rs2.getInt("success"));
        }

        storeList.add(storeVO);
    }
} catch (Exception e) {
    e.printStackTrace();
}
return storeList;
}

public int savefavorite(String user, String code) {
    String SQL = "";

    SQL = "INSERT INTO favorite_store (name, store_code) VALUES (?,?)";

    try {

        pstmt = con.prepareStatement(SQL);
        pstmt.setString(1, user);
        pstmt.setString(2, code);
        return pstmt.executeUpdate();

    } catch (Exception e) {

        e.printStackTrace();

    }

    return -1;
}

public int unsavefavorite(String user, String code) {
    String SQL = "";

    SQL = "DELETE FROM favorite_store WHERE name = ? AND store_code = ?";

    try {

        pstmt = con.prepareStatement(SQL);
        pstmt.setString(1, user);
        pstmt.setString(2, code);
        return pstmt.executeUpdate();

    } catch (Exception e) {

        e.printStackTrace();
    }
}

```

```

    }

    return -1;
}

public ArrayList<StoreVO> favoritelist(String user) {
    String SQL = "", SQL2 = "";
    ArrayList<StoreVO> storeList = new ArrayList<StoreVO>();
    ArrayList<StoreVO> storeList2 = new ArrayList<StoreVO>();
    ArrayList<StoreVO> storeList_sum = new ArrayList<StoreVO>();

    SQL = "SELECT * FROM store ORDER BY store_name ASC";
    try {

        pstmt = con.prepareStatement(SQL);
        rs = pstmt.executeQuery();

        while (rs.next()) {
            StoreVO storeVO = new StoreVO();
            storeVO.setName(rs.getString("store_name"));
            storeVO.setAddress(rs.getString("store_address"));
            storeVO.setCode(rs.getInt("code"));

            SQL2 = "SELECT EXISTS (SELECT * FROM favorite_store
WHERE name = '" + user + "' AND store_code = "
                    + rs.getInt("code") + ") AS success";
            pstmt2 = con2.prepareStatement(SQL2);
            rs2 = pstmt2.executeQuery();

            while (rs2.next()) {
                storeVO.setFavorite(rs2.getInt("success"));
                if (rs2.getInt("success") == 0) {
                    storeList.add(storeVO);
                } else {
                    storeList2.add(storeVO);
                }
            }
        }

        storeList_sum.addAll(storeList2);
        storeList_sum.addAll(storeList);
    } catch (Exception e) {
        e.printStackTrace();
    }
    return storeList_sum;
}

public ArrayList<StoreVO> onlyfavoritelist(String user) {
    String SQL = "", SQL2 = "";
    ArrayList<StoreVO> storeList = new ArrayList<StoreVO>();
    ArrayList<StoreVO> storeList_sum = new ArrayList<StoreVO>();

    SQL = "SELECT * FROM store ORDER BY store_name ASC";
    try {

        pstmt = con.prepareStatement(SQL);
        rs = pstmt.executeQuery();

```

```

        while (rs.next()) {
            StoreVO storeVO = new StoreVO();
            storeVO.setName(rs.getString("store_name"));
            storeVO.setAddress(rs.getString("store_address"));
            storeVO.setCode(rs.getInt("code"));

            SQL2 = "SELECT EXISTS (SELECT * FROM favorite_store
WHERE name = '" + user + "' AND store_code = "
                    + rs.getInt("code") + ") AS success";
            pstmt2 = con2.prepareStatement(SQL2);
            rs2 = pstmt2.executeQuery();

            while (rs2.next()) {
                storeVO.setFavorite(rs2.getInt("success"));
                if (rs2.getInt("success") != 0) {
                    storeList.add(storeVO);
                }
            }
        }

        storeList_sum.addAll(storeList);
    } catch (Exception e) {
        e.printStackTrace();
    }
    return storeList_sum;
}

public StoreVO favoritestore(String user) {
    String SQL = "", SQL2 = "";
    StoreVO storeVO = null;
    SQL = "SELECT * FROM store ORDER BY store_name ASC";
    try {

        pstmt = con.prepareStatement(SQL);
        rs = pstmt.executeQuery();

        while (rs.next()) {
            storeVO = new StoreVO();
            storeVO.setName(rs.getString("store_name"));
            storeVO.setAddress(rs.getString("store_address"));
            storeVO.setCode(rs.getInt("code"));

            SQL2 = "SELECT EXISTS (SELECT * FROM favorite_store
WHERE name = ? AND store_code = ?) AS success";
            pstmt2 = con2.prepareStatement(SQL2);
            pstmt2.setString(1, user);
            pstmt2.setInt(2, storeVO.getCode());
            rs2 = pstmt2.executeQuery();

            while (rs2.next()) {
                storeVO.setFavorite(rs2.getInt("success"));
                if (rs2.getInt("success") == 1) {
                    return storeVO;
                }
            }
        }
    }
}

```



```

        } catch (Exception e) {
            e.printStackTrace();
        }
        return storeVO;
    }

    public ArrayList<StoreVO> searchStore(String search, String user) {
        String SQL = "", SQL2 = "";
        ArrayList<StoreVO> storeList = new ArrayList<StoreVO>();
        ArrayList<StoreVO> storeList_sum = new ArrayList<StoreVO>();

        SQL = "SELECT * FROM store WHERE store_name LIKE ? ORDER BY
store_name ASC";
        try {

            pstmt = con.prepareStatement(SQL);
            pstmt.setString(1, "%" + search + "%");
            rs = pstmt.executeQuery();

            while (rs.next()) {
                StoreVO storeVO = new StoreVO();
                storeVO.setName(rs.getString("store_name"));
                storeVO.setAddress(rs.getString("store_address"));
                storeVO.setCode(rs.getInt("code"));
                if(user != null) {
                    SQL2 = "SELECT EXISTS (SELECT * FROM
favorite_store WHERE name = " + user + " AND store_code = "
+ rs.getInt("code") + ") AS success";
                    pstmt2 = con2.prepareStatement(SQL2);
                    rs2 = pstmt2.executeQuery();

                    while (rs2.next()) {
                        storeVO.setFavorite(rs2.getInt("success"));
                    }
                } else {
                    storeVO.setFavorite(0);
                }
                storeList.add(storeVO);
            }

            storeList_sum.addAll(storeList);
        } catch (Exception e) {
            e.printStackTrace();
        }
        return storeList_sum;
    }
}

```

#### 5.2.4 새로운 물품 등록

##### uploadProduct.jsp

```

<%@page import="java.util.List"%>
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%@page import="com.pjt.cstore.CeoDAO"%>
<!DOCTYPE html>

```

```

<html lang="ko">
<jsp:include page="/include/head.jsp" />
<body>
    <%
        String ceo = (String) session.getAttribute("ceo");
        String user = (String)session.getAttribute("user");

        if (ceo == null) {
            response.sendRedirect("/");
        }

        CeoDAO ceodao = new CeoDAO();
        List<String> storelist = ceodao.getStoreName(user);
    %>
    <div id="page-wrapper">

        <jsp:include page="/include/header.jsp" />

        <jsp:include page="/include/nav.jsp" />

        <section id="main">
            <div class="container">
                <div class="col-12">

                    <div class="content">

                        <article class="box page-content">

                            <header>
                                <h2>물품등록</h2>
                            </header>

                            <section class="box features">
                                <div>
                                    <form
action="/UploadProduct" method="POST" class="form join" id="uploadProductForm"
enctype="multipart/form-data">
                                        <input
type="hidden" name="owner" value="<%=user%>" class="col-6" required />
                                        <div
class="row">
                                            <div class="col-12 row mgt05">

                                                <label for="store" class="col-3">매장 선택</label>

                                                <select id="store" name="store" class="col-6" required >

                                                    <% for(String store : storelist) { %>

                                                        <option><%=store%></option>

                                                    <% } %>

```

```

        </select>
</div>
<div class="col-12 row mgt05">
    <label for="name" class="col-3">상품 이름</label> <input
        type="text" id="name" name="name" class="col-6"
        required />
</div>
<div class="col-12 row mgt05">
    <label for="img" class="col-3">상품 이미지</label>
    <div class="filebox col-9 row mg0 pd0">
        <input class="upload-name col-8" id="img" name="img" value="파일선택"
disabled="disabled" multiple="multiple" required>
        <label class="col-2 mgl50p" for="ex_filename">업로드</label>
        <input type="file" id="ex_filename" name="ex_filename" class="upload-hidden"
multiple="multiple">
    </div>
</div>
<div class="col-12 row mgt05">
    <label for="price" class="col-3">상품 가격</label> <input
        type="text" id="price" name="price" class="col-6"
        required />
</div>
<div class="col-12">
    <ul class="actions">
        <li><a href="javascript:history.back()"
            class="button alt">이전</a> </li>
        <li><button type="button" onClick="submitJoin()"
            class="button">등록완료</button> </li>
    </ul>

```



</html>

### UploadProduct.java

```
package com.pjt.cstore;

import java.io.IOException;

import javax.servlet.ServletContext;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import com.oreilly.servlet.MultipartRequest;
import com.oreilly.servlet.multipart.DefaultFileRenamePolicy;

/**
 * Servlet implementation class UploadProduct
 */
@WebServlet("/UploadProduct")
public class UploadProduct extends HttpServlet {
    private static final long serialVersionUID = 1L;

    /**
     * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
     */
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
        // TODO Auto-generated method stub
        response.getWriter().append("Served at: ").append(request.getContextPath());
    }

    /**
     * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
     */
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
        // TODO Auto-generated method stub
        response.setContentType("text/html;charset=utf-8");
        request.setCharacterEncoding("utf-8");

        ServletContext context = getServletContext();
        String dir = context.getRealPath("assets/img");
        int maxsize = 10*1024*1024; // 10MB
        String encoding = "utf-8";

        MultipartRequest multi = new MultipartRequest(request, dir, maxsize,
encoding, new DefaultFileRenamePolicy());
        String store = multi.getParameter("store");
        String name = multi.getParameter("name");
        String img = multi.getFilesystemName("ex_filename");
        String owner = multi.getParameter("owner");
        int price = Integer.parseInt(multi.getParameter("price"));

        if(!(store.equals("") || name.equals("") || img.equals("") || price==0)) {
```

```

        CeoDAO ceoDAO = new CeoDAO();

        ceoDAO.insertProduct(store,name,img,price,owner);

    }

}

```

## 5.2.5 물품구매 및 장바구니

### basket.jsp

```

<%@page import="java.util.List"%>
<%@page import="java.util.ArrayList"%>
<%@page import="com.pjt.cstore.ProductDAO"%>
<%@page import="com.pjt.cstore.ProductVO"%>
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html lang="ko">

<%
    String user = (String) session.getAttribute("user");

    if (user == null) {
        response.sendRedirect("/");
    }

    ProductDAO productDAO = new ProductDAO();
    ArrayList<ProductVO> productList = null;

    productList = productDAO.basketList(user);
%>

<jsp:include page="/include/head.jsp" />

<body>
    <div id="page-wrapper">

        <jsp:include page="/include/header.jsp" />

        <jsp:include page="/include/nav.jsp" />

        <section id="main">
            <div class="container">
                <form action="/PurchaseController" method="POST"
id="purchaseForm">

                    <div class="row p_relative">
                        <div class="col-3">
                            <div class="sidebar">

                                <!-- Sidebar -->

                                <!-- Recent Posts -->
                            </div>
                        </div>
                    </div>
                </form>
            </div>
        </section>
    </div>
</body>

```



```

        value="<%=product.getIdx()%>" /></td>
<td>" class="basket_img" /><%=product.getName()%>
</td>
<%
        if (product.getSale() != 1) {
%>
<td class="total_price"><%=product.getPrice()%> 원</td>
<%
        } else {
%>
<td class="total_price"><span class="sale
mg0"><%=product.getPrice()%> 원</span><br><%=product.getPrice() -
product.getSaleprice()%> 원</td>
<%
        }
%>
<td class="product_count"><%=product.getCount()%> 개</td>
<%
        if (product.getSale() != 1) {
%>
<td class="sale_price"><%=product.getPrice() * product.getCount()%> 원</td>
<%
        } else {
%>
<td class="sale_price"><%=product.getPrice() - product.getSaleprice() *
product.getCount()%> 원</td>
<%
        }
%>

```



```

        </tr>
        <%
    }
    %>
    <%
    }
    %>
</tbody>
</table>
</div>
</div>
<div class="f_parent w100">
    <button type="button" class="fr
clear_basket mgl50p" onClick="clearbasket()">장바구니
        비우기 </button>
    <button type="button" class="fr
delete_basket" onClick="deletebasket()">선택상품
        삭제 </button>
</div>
<div class="col-12 productDetail">
    <div class="row p_relative mgl50p">
        <table class="mgt50p
basket_2">
<caption>장바구니</caption>
        <colgroup>
            <col
width="33%" />
            <col
width="33%" />
            <col
width="33%" />
        </colgroup>
        <thead>
            <tr>
                <th>총 상품금액</th>
                <th>총 할인금액</th>
                <th>결제예정금액</th>
            </tr>
        </thead>
        <tbody>
            <tr
class="tCenter">
                <td class="sum_total_price">0원</td>
                <td class="sum_sale_price">0원</td>
                <td class="final_price">0원</td>
            </tr>
        </tbody>
    </table>

```

```

        </div>
    </div>
    <div class="col-12">
        <input type="hidden" name="price"
        <input type="hidden" name="type"
        <ul class="actions">
            <li><button type="button"
            <li><button type="button"
        </ul>
    </div>
    </div>
    </form>
</div>
</section>
<!-- Footer 삽입 -->
<jsp:include page="/include/footer.jsp" />
</div>
</body>
<script>
    $(function(){
        var sum_total_price = 0;
        var sum_sale_price = 0;
        var current_sale_price = 0;
        var total_price_text = "";
        var total_sale_price_text = "";
        $(".basket input:checkbox").on("change",function(){
            var product_count = 0;
            product_count =
parseInt($(this).parent().siblings(".product_count").text().split("개")[0]);

            total_price_text =
parseInt($(this).parent().siblings(".total_price").text().split("원")[0]) * product_count;
            if($(this).is(":checked")) {
                sum_total_price += total_price_text;
            } else {
                sum_total_price -= total_price_text;
            }

            $(".sum_total_price").text(sum_total_price+"원");

            total_sale_price_text =
$(this).parent().siblings(".total_price").text().split("원")[1];

            if(total_sale_price_text != "") {

```

```

        current_sale_price = parseInt(total_sale_price_text) *
product_count;
    } else {
        current_sale_price = total_price_text;
    }

    if($(this).is(":checked")) {
        sum_sale_price += current_sale_price;
    } else {
        sum_sale_price -= current_sale_price;
    }

    $(".sum_sale_price").text("-"+(sum_total_price-sum_sale_price)+"원");

    $(".final_price").text(sum_sale_price+"원");
});

$(".purchaseall").on("click", function() {
    $("input[name=checkItem]").each(function() {
        if!$(this).prop("checked") {
            $(this).prop("checked", true).change();
        }
    });
    $("input[name=price]").val($(".final_price").text().split("원")[0]);
    $("#purchaseForm").submit();
});

$(".purchaseselect").on("click", function() {
    if($(".input[name=checkItem]:checked").length == 0) {
        alert("선택된 상품이 없습니다.");
    } else {
        $("input[name=price]").val($(".final_price").text().split("원")[0]);
        $("#purchaseForm").submit();
    }
});

});

function deletebasket() {
    if($(".basket input:checkbox:checked").length == 0) {
        alert("선택된 항목이 없습니다.");
    } else {
        var pcode="";
        for(var i=0; i < $(".basket input:checkbox:checked").length; i++) {
            pcode += $(".basket input:checkbox:checked").eq(i).attr('id') +
";";
        }

        $.ajax({
            async : true,
            type : 'GET',
            url :
"/ProductController?id=<%=user%>&type=deleteProduct&pcode=" + pcode,
            dataType : "text",
            contentType : "application/json; charset=UTF-8",
            success : function(data) {

```

```

        if(data == "success") {
            alert("선택한 상품을 장바구니에서
삭제했습니다.");
            location.reload();
        } else {
            alert("선택한 상품 삭제에 실패했습니다.");
        }
    }
});
}
}
}

function clearbasket() {
    $.ajax({
        async : true,
        type : 'GET',
        url : "/ProductController?id=<%=user%>&type=clearBasket",
        dataType : "text",
        contentType : "application/json; charset=UTF-8",
        success : function(data) {
            if (data == "success") {
                alert("장바구니를 비웠습니다.");
                location.reload();
            } else {
                alert("장바구니를 비우기에 실패했습니다.");
            }
        }
    });
}
}
</script>
</html>

```

### purchase.jsp

```

<%@page import="com.pjt.cstore.PurchaseVO"%>
<%@page import="com.pjt.cstore.PurchaseDAO"%>
<%@page import="java.util.List"%>
<%@page import="java.util.ArrayList"%>
<%@page import="com.pjt.cstore.ProductDAO"%>
<%@page import="com.pjt.cstore.ProductVO"%>
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html lang="ko">

<%
    String user = (String) session.getAttribute("user");

    if (user == null) {
        response.sendRedirect("/");
    }

    ProductDAO productDAO = new ProductDAO();
    PurchaseDAO purchaseDAO = new PurchaseDAO();

    ArrayList<PurchaseVO> purchaseList = null;
    ArrayList<ProductVO> productList = null;

```

```

        purchaseList = purchaseDAO.getList(user);
        productList = productDAO.purchaseList(user,purchaseList);
%>
<jsp:include page="/include/head.jsp" />
<body>
    <div id="page-wrapper">
        <jsp:include page="/include/header.jsp" />
        <jsp:include page="/include/nav.jsp" />
        <section id="main">
            <div class="container">
                <form action="/PurchaseController" method="POST"
id="purchaseForm">
                    <div class="row p_relative">
                        <div class="col-3">
                            <div class="sidebar">
                                <section>
                                    <h2 class="major">
<span>구매하기</span>
                                </h2>
                                </section>
                            </div>
                        </div>
                        <div class="col-12 productDetail">
                            <div class="row p_relative"
borderBottom borderTop mgl50p">
                                <table class="mgt50p"
basket">
                                    <caption>구매하기</caption>
                                    <colgroup>
                                        <col
width="20%" />
                                        <col
width="*" />
                                        <col
width="15%" />
                                        <col
width="10%" />
                                        <col
width="15%" />
                                    </colgroup>
                                    <thead>
                                        <tr>
<th>점포명</th>
<th>상품명</th>

```

```

<th>가격</th>
<th>수량</th>
<th>합계</th>
</tr>
</thead>
<tbody>
<%
if
(productList.size() != 0) {
%>
%>
for
(ProductVO product : productList) {
%>
<tr
class="tCenter">
<td> <input type="checkbox" name="checkItem"
class="chkbox" id="<%=product.getIdx()%>"
value="<%=product.getIdx()%>" /> </td>
<td> " class="basket_img" /> <%=product.getName()%>
</td>
<%
if (product.getSale() != 1) {
%>
<td class="total_price"> <%=product.getPrice()%> 원 </td>
<%
} else {
%>
<td class="total_price"> <span class="sale
mg0"> <%=product.getPrice()%> 원 </span> <br> <%=product.getPrice() -
product.getSaleprice()%> 원 </td>
<%
}
%>
<td class="product_count"> <%=product.getCount()%> 개 </td>

```

```

<%
    if (product.getSale() != 1) {
%>
<td class="sale_price"><%=product.getPrice() * product.getCount()%>원</td>
<%
    } else {
%>
<td class="sale_price"><%=(product.getPrice() - product.getSaleprice()) *
product.getCount()%>원</td>
<%
    }
%>
</tr>
<%
    }
%>
</tbody>
</table>
</div>
</div>
<div class="f_parent w100">
    <button type="button" class="fr
clear_basket mgl50p" onClick="resetpurchase()">이전페이지(구매목록 삭제)</button>
    <button type="button" class="fr
delete_basket" onClick="deletepurchase()">선택상품
        삭제</button>
</div>
<div class="col-12 productDetail">
    <div class="row p_relative mgl50p">
        <table class="mgt50p
basket_2">
<caption>구매하기</caption>
        <colgroup>
            <col
width="33%" />
            <col
width="33%" />
            <col
width="33%" />
        </colgroup>

```

```

<thead>
  <tr>
    <th>총 상품금액</th>
    <th>총 할인금액</th>
    <th>결제예정금액</th>
  </tr>
</thead>
<tbody>
  <tr>
    class="tCenter">
      <td class="sum_total_price">0원</td>
      <td class="sum_sale_price">0원</td>
      <td class="final_price">0원</td>
    </tr>
  </tbody>
</table>
</div>
<div class="col-12">
  <input type="hidden" name="price"
value="0">
  <input type="hidden" name="type"
value="purchase">
  <input type="hidden" name="id"
value="<%=user%>">
  <ul class="actions">
    <li><button type="button"
class="button large purchaseall">전체
상품구매</button></li>
    <li><button type="button"
class="button alt large purchaseselect">선택 상품구매</button></li>
  </ul>
</div>
</div>
</form>
</div>
</section>
<jsp:include page="/include/footer.jsp" />
</div>
</body>
<script>
  $(function(){
    var sum_total_price = 0;
    var sum_sale_price = 0;

```



```

var current_sale_price = 0;
var total_price_text = "";
var total_sale_price_text = "";
$(".basket input:checkbox").on("change",function(){
    var product_count = 0;
    product_count =
parseInt($(this).parent().siblings(".product_count").text().split("개")[0]);

    total_price_text =
parseInt($(this).parent().siblings(".total_price").text().split("원")[0]) * product_count;
    if($(this).is(":checked")) {
        sum_total_price += total_price_text;
    } else {
        sum_total_price -= total_price_text;
    }

    $(".sum_total_price").text(sum_total_price+"원");

    total_sale_price_text =
$(this).parent().siblings(".total_price").text().split("원")[1];

    if(total_sale_price_text != "") {
        current_sale_price = parseInt(total_sale_price_text) *
product_count;
    } else {
        current_sale_price = total_price_text;
    }

    if($(this).is(":checked")) {
        sum_sale_price += current_sale_price;
    } else {
        sum_sale_price -= current_sale_price;
    }

    $(".sum_sale_price").text("-"+(sum_total_price-sum_sale_price)+"원");

    $(".final_price").text(sum_sale_price+"원");
});

$(".purchaseall").on("click", function() {
    $("input[name=checkItem]").each(function() {
        if(!$(this).prop("checked")) {
            $(this).prop("checked", true).change();
        }
    });
    $("input[name=price]").val($(".final_price").text().split("원")[0]);
    $("#purchaseForm").submit();
});

$(".purchaseselect").on("click", function() {
    if($(".input[name=checkItem]:checked").length == 0) {
        alert("선택된 상품이 없습니다.");
    } else {
        $("input[name=price]").val($(".final_price").text().split("원")[0]);
        $("#purchaseForm").submit();
    }
});

```

```

    }
  })
});

function deletepurchase() {
  if($(".basket input:checkbox:checked").length == 0) {
    alert("선택된 항목이 없습니다.");
  } else {
    var pcode="";
    for(var i=0; i < $(".basket input:checkbox:checked").length; i++) {
      pcode += $(".basket input:checkbox:checked").eq(i).attr('id') +
";";
    }

    $.ajax({
      async : true,
      type : 'GET',
      url :
"/ProductController?id=<%=user%>&type=deletepurchase&pcode=" + pcode,
      dataType : "text",
      contentType : "application/json; charset=UTF-8",
      success : function(data) {
        if(data == "success") {
          alert("선택한 상품을 구매목록에서
삭제했습니다.");
          location.reload();
        } else {
          alert("선택한 상품 삭제에 실패했습니다.");
        }
      }
    });
  }
}

function resetpurchase() {
  $.ajax({
    async : false,
    type : 'GET',
    url : "/ProductController?id=<%=user%>&type=reset",
    dataType : "text",
    contentType : "application/json; charset=UTF-8",
    success : function(data) {
      if(data == "success") {
        history.go(-1);
      }
    }
  });
}
}
</script>
</html>

```