
PORTFOLIO

CONTENTS

- 1 기획 의도
- 2 개발 환경
- 3 요구사항 정의서
- 4 UI 설계
- 5 DB 설계
- 6 주요 기술 상세

Part 1, 기획 의도





편안한 제품을 고르기 위해

베스트 상품을 노출시켜 물건을 고르기 힘든 분들에게 도움이 되는 방향으로 제작했습니다



안전한 제품을 고르기 위해

여러 인증제도를 통과해 기저귀로 아이가 해를 입지 않는 안전하고 믿을 수 있는 기저귀라는 것을 알릴 수 있도록 제작했습니다



함께할 제품을 고르기 위해

리뷰, 고객지원 등으로 제품 구매 이후에도 고객님들이 만족감을 느낄 수 있게 제작하였습니다.

Part 2, 개발 환경



Software

- ✓ JAVA 1.8
- ✓ Eclipse
- ✓ Apache Tomcat 8.5
- ✓ MySQL (Workbench 8.0 CE)
- ✓ Spring Framework
- ✓ HTML5
- ✓ CSS
- ✓ JQuery

Server Computer

- ✓ OS : Window 10
- ✓ CPU : CPU : Intel(R) Core(™) i5-4690 CPU @ 3.50GHz
- ✓ RAM : 8 GB
- ✓ SSD : 250GB

Client Computer

- ✓ OS : Window 10
- ✓ CPU : CPU : Intel(R) Core(™) i5-4690 CPU @ 3.50GHz
- ✓ RAM : 8 GB
- ✓ SSD : 250GB

Part 3, 요구사항 정의서



요구사항 정의서

	이름	내용	우선순위	중요도
로그인	로그인	로그인 페이지로 이동한다	필수	상
	로그인 확인	로그인 하려는 사용자가 DB에 존재하는지 확인한다.	필수	상
회원가입	회원가입	회원가입 페이지로 이동한다.	필수	상
	아이디 유효성 검사	아이디가 형식에 맞는지 확인한다	필수	상
	비밀번호 유효성 검사	비밀번호가 형식에 맞는지 확인한다	필수	상
	비밀번호 일치 확인	비밀번호가 일치하는지 확인한다	필수	상
	이메일 유효성 검사	이메일이 형식에 맞는지 확인한다	필수	상
	휴대전화 번호 유효성 검사	휴대번호가 형식에 맞는지 확인한다	필수	상
	회원가입	입력한 내용을 토대로 회원가입을 한다. 입력한 정보는 DB에 저장된다.	필수	상
헤더	로그인 여부	로그인이 성공했을 경우 메인 헤더의 메뉴에 인사말 / 로그아웃' 기능이 생성된다.	필수	상
	메인 로고	로고를 누르면 맨 처음 메인 페이지로 이동한다.	필수	상
	제품별리스트	제품별리스트 페이지로 이동한다.	필수	상
	고객지원	고객지원 페이지로 이동한다.	필수	상

요구사항 정의서

	이름	내용	우선순위	중요도
상세 페이지	상세 페이지	상세 페이지로 이동한다	필수	상
댓글	권한 확인	관리자 아이디로 로그인 되었을 때만 댓글 작업 가능	필수	상
	댓글 작성	내용이 작성되었는지 확인한다.	필수	상
	댓글 수정	수정 버튼을 클릭할 시 모달 창을 띄운다	필수	상
	댓글 삭제	삭제 버튼을 클릭할 시 해당 댓글을 삭제한다	필수	상
글 수정	권한 확인	글 수정 & 삭제하려는 사용자가 작성자와 일치하는지 확인	필수	상
	글 수정	수정 화면으로 이동한다	필수	상
	글 삭제	글이 삭제되고 목록 페이지로 이동한다	필수	상

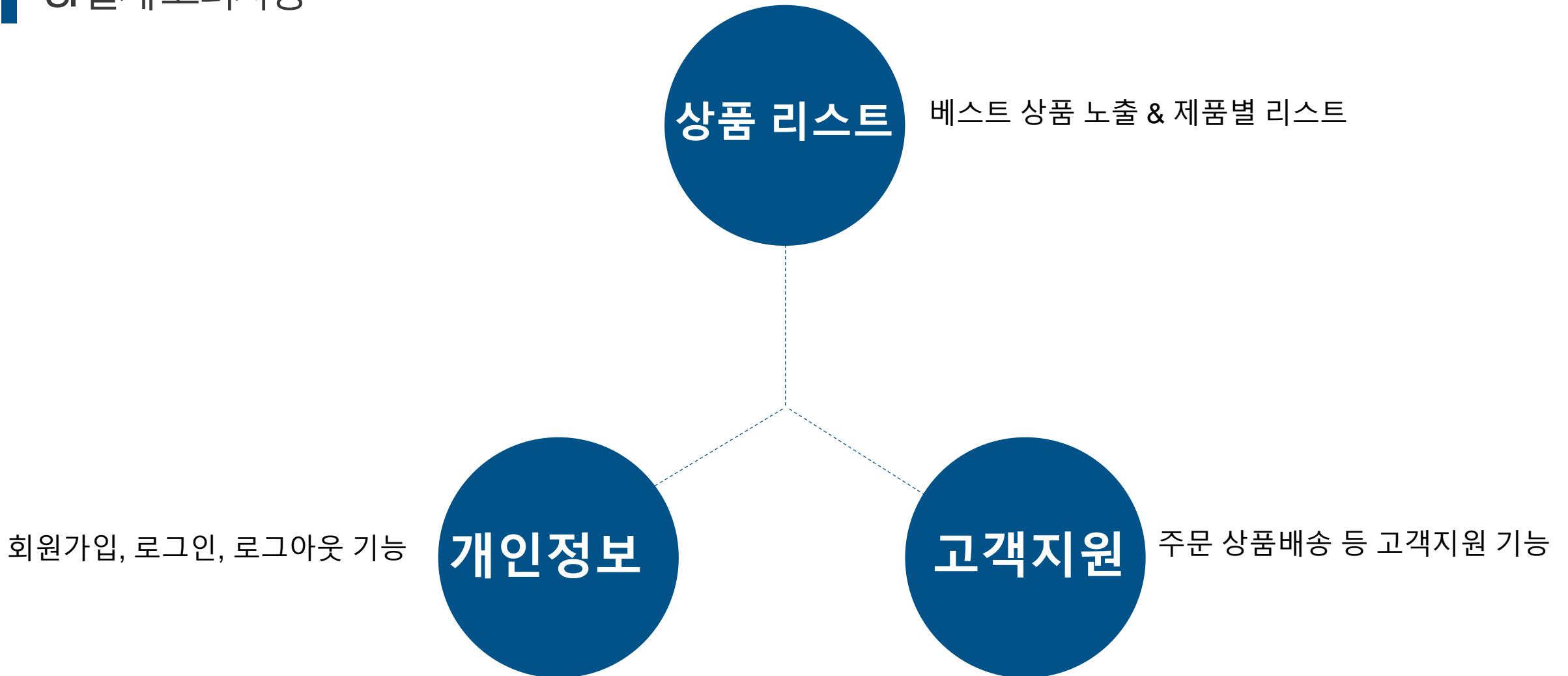
요구사항 정의서

	이름	내용	우선순위	중요도
고객 센터	로그인 여부	로그인시 글쓰기 버튼이 활성화 된다.	필수	상
	글쓰기 페이지	글쓰기 버튼을 누를 시 페이지를 이동한다	필수	상
	검색	고객 문의에 대한 정보(작성자, 제목)을 검색할 시 해당 목록을 첫번째 페이지에 띄운다.	필수	중
제품 리스트	권한 확인	관리자로 로그인시 상품 추가 버튼 활성화	필수	상
	검색	제품에 대한 정보(제품 이름, 가격)을 검색할 시 해당 목록을 첫번째 페이지에 띄운다.	필수	중
	제품 정렬	제품 정렬에 대한 옵션 변경에 맞게 정렬된다.	필수	중

Part 4, UI 설계



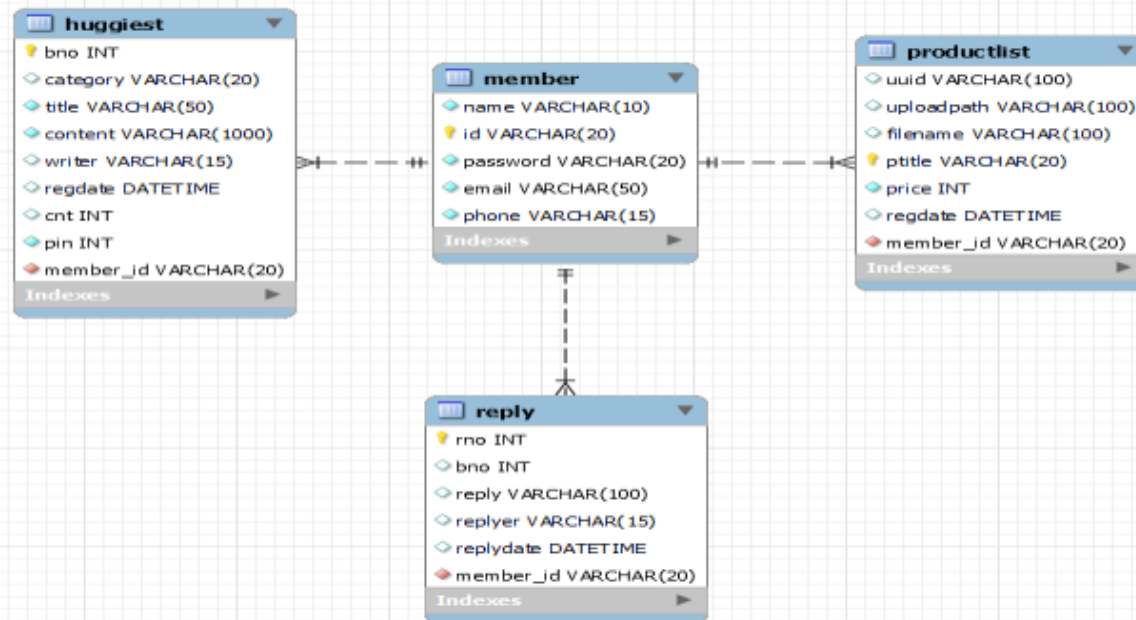
UI 설계 고려사항



Part 5, DB 설계



ERD



MEMBER 테이블 명세

Column Name	Datatype	PK	Not Null	설명
name	VARCHAR(10)		○	이름
id	VARCHAR(20)	○	○	아이디
password	VARCHAR(20)		○	비밀번호
email	VARCHAR(50)		○	이메일
phone	VARCHAR(15)		○	휴대전화 번호

HUGGIEST 테이블 명세

Column Name	Datatype	PK	Not Null	설명
bno	INT	○	○	게시글 번호
category	VARCHAR(20)			카테고리
title	VARCHAR(50)		○	제목
content	VARCHAR(1000)		○	내용
writer	VARCHAR(15)			작성자
regdate	DATETIME			작성 날짜
cnt	INT			조회수
pin	INT		○	공지사항 고정

REPLY 테이블 명세

Column Name	Datatype	PK	Not Null	설명
rno	INT	○	○	댓글 번호
bno	INT		○	게시글 번호
reply	VARCHAR(1000)			댓글 내용
replyer	VARCHAR(10)			댓글 작성자
replydate	DATETIME			댓글 작성 날짜

PRODUCT 테이블 명세

Column Name	Datatype	PK	Not Null	설명
UUID	VARCHAR(100)			UUID
uploadPath	VARCHAR(100)			파일 경로
fileName	VARCHAR(100)			파일 이름
ptitle	VARCHAR(20)	○		제품명
price	Int		○	가격
regdate	DATETIME			작성 날짜

Part 6, 주요 기능 상세



Github

<https://github.com/greenTeam4/huggies.git>

회원가입

```
</div>
<form action="/join" method="post">
```

Join.jsp

```
@PostMapping("join")
public String postjoin(MemberDTO mdto){
    Mservice.insert(mdto);
    return "redirect:index";
}
```

memberController

```
1 package org.huggies.service;
2
3 import org.huggies.domain.MemberDTO;
4
5 public interface Memberservice {
6
7     public void insert(MemberDTO mdto);
8
9     public MemberDTO login(MemberDTO mdto);
10
11 }
12
```

memberService

```
<mapper namespace="org.huggies.mapper.MemberMapper">
<insert id="insert">
    insert into member
    values(#{name}, #{id}, #{password}, #{email}, #{phone})
</insert>
```

memberMapper

```
1 package org.huggies.mapper;
2
3 import org.huggies.domain.MemberDTO;
4
5 public interface MemberMapper {
6
7     public void insert(MemberDTO mdto);
8
9     public MemberDTO login(MemberDTO mdto);
10
11 }
12
```

memberMapper

```
1 package org.huggies.service;
2
3 import org.huggies.domain.MemberDTO;
4
5 @Service
6 public class Memberserviceimpl implements Memberservice{
7
8     @Autowired
9     private MemberMapper mmapper;
10
11     public void insert(MemberDTO mdto) {
12         mmapper.insert(mdto);
13     }
14
15     public MemberDTO login(MemberDTO mdto) {
16         return mmapper.login(mdto);
17     }
18
19 }
20
21
22
23
```

memberServiceImp

로그인

```
<form action="/login" method="post">
```

login.jsp

```
@PostMapping("login")
public String Postlogin(MemberDTO mdto, HttpSession session) {
    MemberDTO login = Mservice.login(mdto);

    session.setAttribute("login", login);

    if(session.getAttribute("login")!=null) {
        return "redirect:/index";
    }else{
        return "redirect:/login";
    }
};
```

memberController

```
1 package org.huggies.service;
2
3 import org.huggies.domain.MemberDTO;
4
5 public interface Mberservice {
6
7     public void insert(MemberDTO mdto);
8
9     public MemberDTO login(MemberDTO mdto);
10 }
11
12
```

memberService

```
<select id="Login" resultType="org.huggies.domain.MemberDTO">
    select *
    from member
    where id=#{id}
    and password=#{password}
</select>
```

memberMapper

```
1 package org.huggies.mapper;
2
3 import org.huggies.domain.MemberDTO;
4
5 public interface MemberMapper {
6
7     public void insert(MemberDTO mdto);
8
9     public MemberDTO login(MemberDTO mdto);
10 }
11
12
```

memberMapper

```
1 package org.huggies.service;
2
3 import org.huggies.domain.MemberDTO;
4
5 @Service
6 public class Mberserviceimpl implements Mberservice{
7
8     @Autowired
9     private MemberMapper mmapper;
10
11     public void insert(MemberDTO mdto) {
12         mmapper.insert(mdto);
13     }
14
15     public MemberDTO login(MemberDTO mdto) {
16         return mmapper.login(mdto);
17     }
18 }
19
20
21
22
23
```

memberServiceImp

마이페이지

```
<a href="/mypage">마이페이지</a>
```

Header.jsp

```
@GetMapping("mypage")
public String mypage(HttpSession session){
    if(session.getAttribute("login")!=null) {
        return "mypage";
    }else{
        return "redirect:/login";
    }
};
```

memberController

```
<c:if test="${login!=null}"><p>${login.id}님 안녕하세요</p></c:if>
</div>
<ul>
<li class="ul-f"><a href="#">나의 쇼핑</a></li>
<div class="horizon"></div>
<li class="ul-f"><a href="#">관심 상품</a></li>
<ul>
<li class="uls"><a href="#">장바구니 상품</a></li>
<li class="uls-b"><a href="#">최근 본 상품</a></li>
</ul>
<div class="horizon"></div>
```

Mypage.jsp

게시판 상세 페이지1 (상세 페이지로 이동)

```
<a href="/detail?bno=${board.bno}">${board.title}</a>
```

List.jsp

```
// <게시판 상세 페이지>로 이동  
@GetMapping("detail")  
public void detail(BoardDTO board, Model model) {  
    model.addAttribute("detail", bservice.detail(board));  
}
```

HuggiesController

```
// 게시판 상세 페이지 설계  
public BoardDTO detail(BoardDTO board);
```

BoardService



```
<update id="cntupdate">  
    update huggiest  
    set cnt = cnt+1  
    where bno = #{bno}  
</update>  
  
<select id="detail" resultType="org.huggies.domain.BoardDTO">  
    select *  
    from huggiest  
    where bno=#{bno}  
</select>
```

BoardMapper

```
// 게시판 목록리스트에서 제목을 클릭했을때 조회수 +1  
public void cntupdate(BoardDTO board);  
// 게시판 상세 페이지 설계  
public BoardDTO detail(BoardDTO board);
```

BoardMapper

```
@Transactional  
public BoardDTO detail(BoardDTO board) {  
    bmapper.cntupdate(board);  
    return bmapper.detail(board);  
}
```

BoardService
Impl

Part 6, 주요 기능 상세

게시판 상세 페이지2(글 수정&글 삭제)

```
<div class="board_content">
  ${detail.content}
  <c:if test="${detail.writer == login.name}">
    <div class="board_btns">
      <a href="/modify?bno=${detail.bno}">글 수정</a>
      <a href="/remove?bno=${detail.bno}">글 삭제</a>
    </div>
  </c:if>
</div>
```

Detail.jsp

```
// <게시판 수정 페이지>로 이동
@GetMapping("modify")
public void modify(BoardDTO board, Model model) {
    model.addAttribute("detail", bservice.detail(board));
}
```

HuggiesController

```
<c:when test="${detail.category eq '공지사항'}">
  <select name="category">
    <option value="notice" selected>공지사항 (관리자만 선택 가능합니다) </option>
    <option value="order/pay">주문/결제 </option>
    <option value="delivery">배송 </option>
    <option value="refund/exchange/return">환불/교환/반품 </option>
    <option value="goods">상품 </option>
  </select>
</c:when>
```

Modify.jsp

```
<div class="board_content">
  ${detail.content}
  <c:if test="${detail.writer == login.name}">
    <div class="board_btns">
      <a href="/modify?bno=${detail.bno}">글 수정</a>
      <a href="/remove?bno=${detail.bno}">글 삭제</a>
    </div>
  </c:if>
</div>
```

Detail.jsp

```
// <게시판 상세 페이지>에서 삭제 버튼을 클릭하면
@GetMapping("remove")
public String remove(BoardDTO board) {
    bservice.remove(board);

    return "redirect:/list";
}
```

HuggiesController

```
<%@include file="header.jsp" %>

<div id="board_contents">
  <div class="sub_title">
    <h1 class="st">
      <a href="/list">고객 지원</a>
    </h1>
  </div>
</div>
```

List.jsp

게시판 상세 페이지3(댓글)

```
<!-- 댓글 작성 -->
<div class="board_reply">
    <input id="bno" type="hidden" value="${detail.bno}" />
    <!-- if test: ${login.id == 'administrator'} -->
    <div>댓글</div>
    <div>
        <textarea id="reply" maxlength="2000" cols="125" rows="3" required>
        <div class="board_reply_bottom">
            <label for="replyer">작성자</label>
            <input id="replyer" type="text" value="${login.name}" />
            <button id="addReply">댓글 작성</button>
        </div>
    </div>
</div>
</div>
<div>
    <input id="session" type="hidden" value="${login.id}" />
    <ul id="relist">
    </ul>
</div>
</div>
```

Detail.jsp

```
// 댓글 쓰기 함수 선언
function add(reply,callback) {
    $.ajax({
        url:"/replies/new",
        type:"post",
        data:JSON.stringify(reply),
        contentType : "application/json;charset=UTF-8",
        success: function(result){
            if(callback)
                callback(result);
        }
    })
}
```

Detail.js

```
@PostMapping(value="/new", consumes="application/json", produces={MediaType.TEXT_PLAIN_VALUE})
public ResponseEntity<String> create(@RequestBody ReplyDTO rdto) {
    List result = rservice.write(rdto);
    return result==?new ResponseEntity<>("success",HttpStatus.OK)
        :new ResponseEntity<>(HttpStatus.INTERNAL_SERVER_ERROR);
}
```

ReplyController

```
//댓글작성, 댓글작성 댓글 내용물 가져오기
let reply = $("#reply").val();
//댓글작성, 댓글작성 댓글 작성자를 가져오기
let replyer = $("#replyer").val();

replyService.add({reply:reply, replyer:replyer, bno:bno},
    function(result){
        alert("댓글 작성 : " + result);
        showList();
    }
);
$("#reply").val("");
$("#replyer").val("");
```

Detail.js

```
// 댓글 쓰기 설계
public int write(ReplyDTO rdto);

<insert id="write">
    insert into reply(bno,reply,replyer)
    values(#{bno},#{reply},#{replyer})
</insert>
```

ReplyMapper

```
// 댓글 쓰기 설계
public int write(ReplyDTO rdto);

// 댓글 쓰기 설계한 것을 구현
public int write(ReplyDTO rdto) {
    return rmapper.write(rdto);
}
```

ReplyService

주요 기능 상세

페이징&검색

[illegible]

List.jsp

```
// <게시판 목록 페이지>트 이음
@GetMapping("list")
public void list(Criteria cri, Model model) {
    model.addAttribute("list", service.list(cri));
    int total = service.getTotalCount(cri);
    model.addAttribute("pageMaker", new pageDTO(cri, total));
}
```

HuggiesController

```
// 게시판 페이징에 쓰일 페이징 전체
public int getTotalCount(Criteria cri);
```

BoardService

```
select id, 'let' resulttype, 'org.hugoboss.chess.chessboard' boardid
  select *
  from(
select @rownum:=@rownum+1 as rownum, ht.*
  from (select @rownum:=0) as tmp,
       hugoboss as ht
  <# test='keyword' is null?
   where title like concat('%',#{keyword},'%')
   or content like concat('%',#{keyword},'%')
   or writer like concat('%',#{keyword},'%')
  )#
  order by pin desc, regdate desc
  ) as boardlist
<![CDATA[
  where rownum > #{@pageNum-1}*#{@amount}
  and rownum <= #{@pageNum}*#{@amount}
]]>
```

BoardMapper

```
// 게시판 페이징에 쓰일 페이징 전체
public int getTotalCount(Criteria cri);
```

BoardMapper

```
// 게시판 페이지에 쓰일 페이지 전체
public int getTotalCount(Criteria cri) {
    return bmapper.getTotalCount(cri);
}
```

BoardService
Impl [©]

주요 기능 상세

이미지 파일 업로드

[illegible]

addProduct.jsp

```

41
```

addProduct.js

```
g = g + 1;
for (int i = 0; i < n; i++)
    g[i] = g[i] + 1;
return g;
```

ProductList.jsp

```

1  #!/usr/bin/perl
2
3  use strict;
4  use warnings;
5
6  # Import the Net::FTP module
7  use Net::FTP;
8
9  # Create a new Net::FTP object
10 my $ftp = Net::FTP->new('192.168.1.1');
11
12 # Connect to the FTP server
13 $ftp->connect();
14
15 # Login with username and password
16 $ftp->login('user', 'password');
17
18 # List the files in the current directory
19 $ftp->ls();
20
21 # Get the file size of the file 'example.txt'
22 my $size = $ftp->size('example.txt');
23
24 # Print the file size
25 print "File size: $size\n";
26
27 # Close the connection
28 $ftp->quit();
29
30 # End of script
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

```

ProductController

제품별 리스트(상품 추가페이지)

```
<div class="product_main">
  <c:if test="${login.id == 'administrator1'}">
    <div class="Pmore">
      <a href="/addProduct">
        <button class="btn_Pmore">상품추가</button>
      </a>
    </div>
  </c:if>
</div>
```

productList.jsp



```
// <제품 등록 페이지>에서 등록 버튼을 클릭하면
@PostMapping("addProduct")
public String addProductPost(ProductDTO2 product) {
    pservice.addProduct(product);

    return "redirect:/product";
}
```

ProductController



```
<div id="ap_box">
  <h2>상품 등록</h2>

  <div class="ap_in">
    <form role="form" class="ap_form" action="/addProduct" method="post">
      <div class="ap_row">
        <div class="ap_left">
          <p>상품명</p>
        </div>
        <div class="ap_right">
          <input type="text" name="ptitle" required>
        </div>
      </div>
    </form>
  </div>
</div>
```

addProduct.jsp

「*감사합니다*」