
PORTFOLIO

목차

a table of contents

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



Part 1

기획 의도



각양각색 '커스텀 케이크' 전성시대

나건웅 기자 · 입력 : 2020.10.07 10:04:41 · 최종수정 : 2020.10.09 09:18:01

산 가 순

서울 강서구에 사는 직장인 김재민 씨는 최근 '환갑을 맞은 모친 생일 선물로 '케이크'를 선택했다. '너무 익숙한 것 아니냐'는 생각이 들 수 있지만 오산. 김 씨가 주문한 케이크 가격은 무려 160만원이다. 오히려 친구들이 좋아하는 꽃 종류만 골라 '양금플라워떡케이크'를 주문제작했다. 케이크 위에 피어난 정교한 꽃 장식에 주인공의 표정은 이미 상글병글. 하지만 진정한 하이라이트는 그 다음부터다. '○○○ 씨 오래오래 사세요'라고 적힌 투파를 들어 올리면 현금이 들어 있는 돈 주머니가 줄줄이 따라 나온다. '반천 케이크'에 그야말로 분위기는 잔칫집이 된다. 김 씨는 "먹기가 아까울 정도로 케이크가 너무 예뻐다. 특수 제작한 케이크에서 돈다발이 나오자 입을 다물지 못하시는 어머니 모습에 마음이 너무 흐뭇했다"며 미소 지었다.

코로나 확진자 늘자...안전한 '홈파티'가 대세

한소진 기자·김채원 연필기자 sjin@hankooki.com · 촬영 2021.12.08 12:30



출처
나건웅 기자
wasabi@mk.co.kr]

[본 기사는 매경이코
노미 제2078호
(2020.10.07~10.13일
자) 기사입니다]

■ 레터링 케이크의 수요 증가

코로나로 인해 사람들이 생일과 기념
일 날 소수의 사람들과 모여 노는
'홈 파티'유행

-> 특별한 sns인증 사진을 위해
주문 제작한 케이크가 유행

출처
기자천소진 기자
· 김채원연기자

본 기사의 저작권은
한국미디어네트워
크에 있습니다. 무단전
재 및 재배포 금지

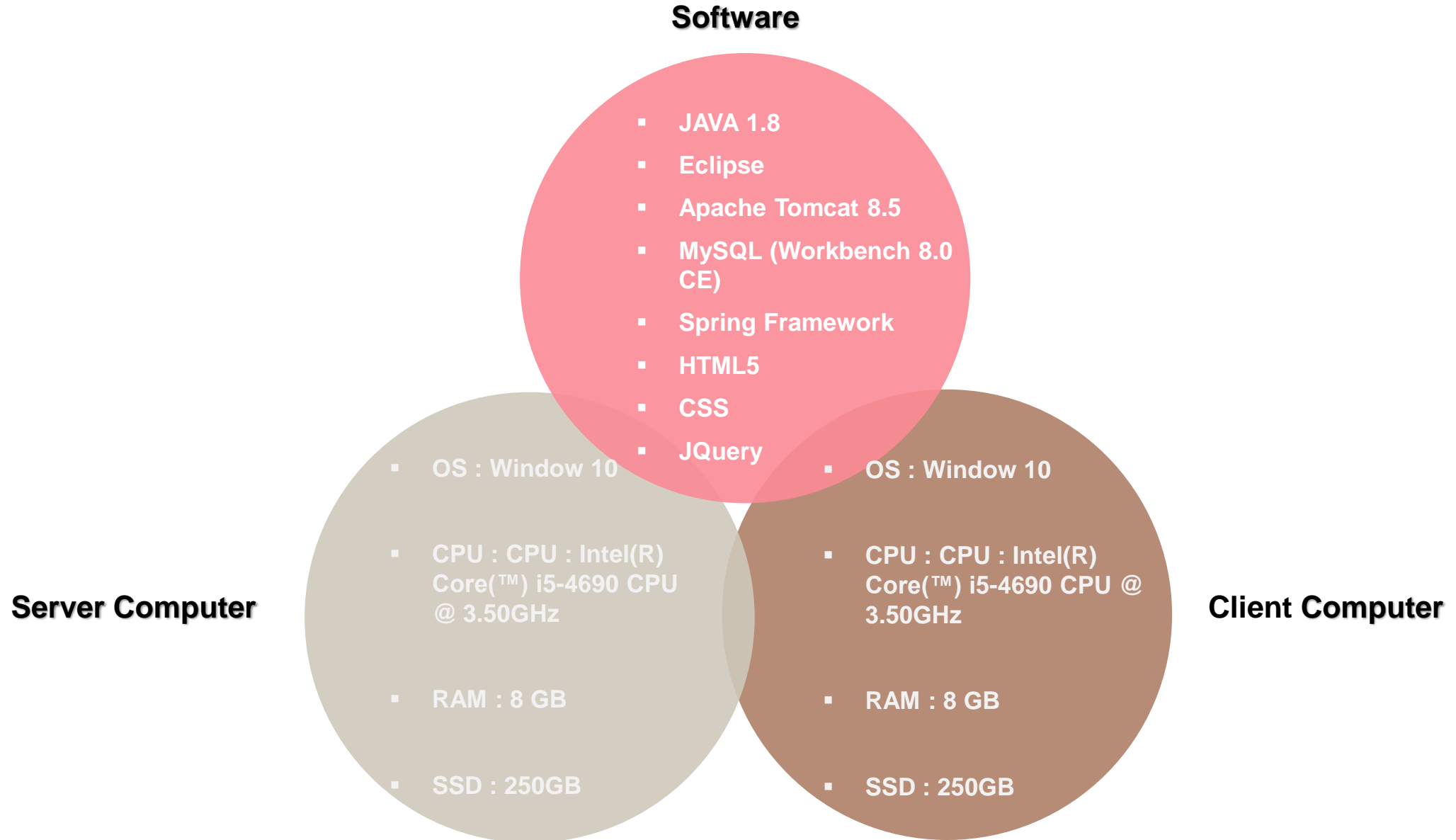
■ 다양한 연령층 공략

대부분 SNS를 통해 주문하여 해당
SNS를 하지않거나, 이러한 주문 방
식이 익숙한 소비자들이 보다 익숙
한 방식으로 주문 할 수 있도록 돕기
위하여 기획

Part 2

개발 환경





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	중
	비밀번호 유효성 검사	비밀번호가 형식에 맞는지 확인한다	필수	중
	비밀번호 일치 확인	비밀번호가 일치하는지 확인한다	필수	중
	이메일 유효성 검사	이메일이 형식에 맞는지 확인한다	필수	중
	휴대전화 번호 유효성 검사	휴대번호가 형식에 맞는지 확인한다	필수	중
	회원가입	입력한 내용을 토대로 회원가입을 한다. 입력한 정보는 DB에 저장된다.	필수	상
	로그인 여부	로그인이 성공했을 경우 메인 헤더의 메뉴에 '로그아웃' 기능이 생성된다.	필수	중
헤더	메인 로고	로고를 누르면 맨 처음 메인 페이지로 이동한다.	필수	하
	제품별리스트	제품별리스트 페이지로 이동한다.	필수	상
	고객지원	고객지원 페이지로 이동한다.	필수	상

요구사항 정의서

	이름	내용	우선순위	중요도
COMMUNITY	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
cake	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
명예의 전당	로그인 여부	로그인시 리뷰 버튼이 활성화 된다.	필수	상
	글쓰기 페이지	글쓰기 버튼을 누를 시 페이지를 이동한다	필수	상

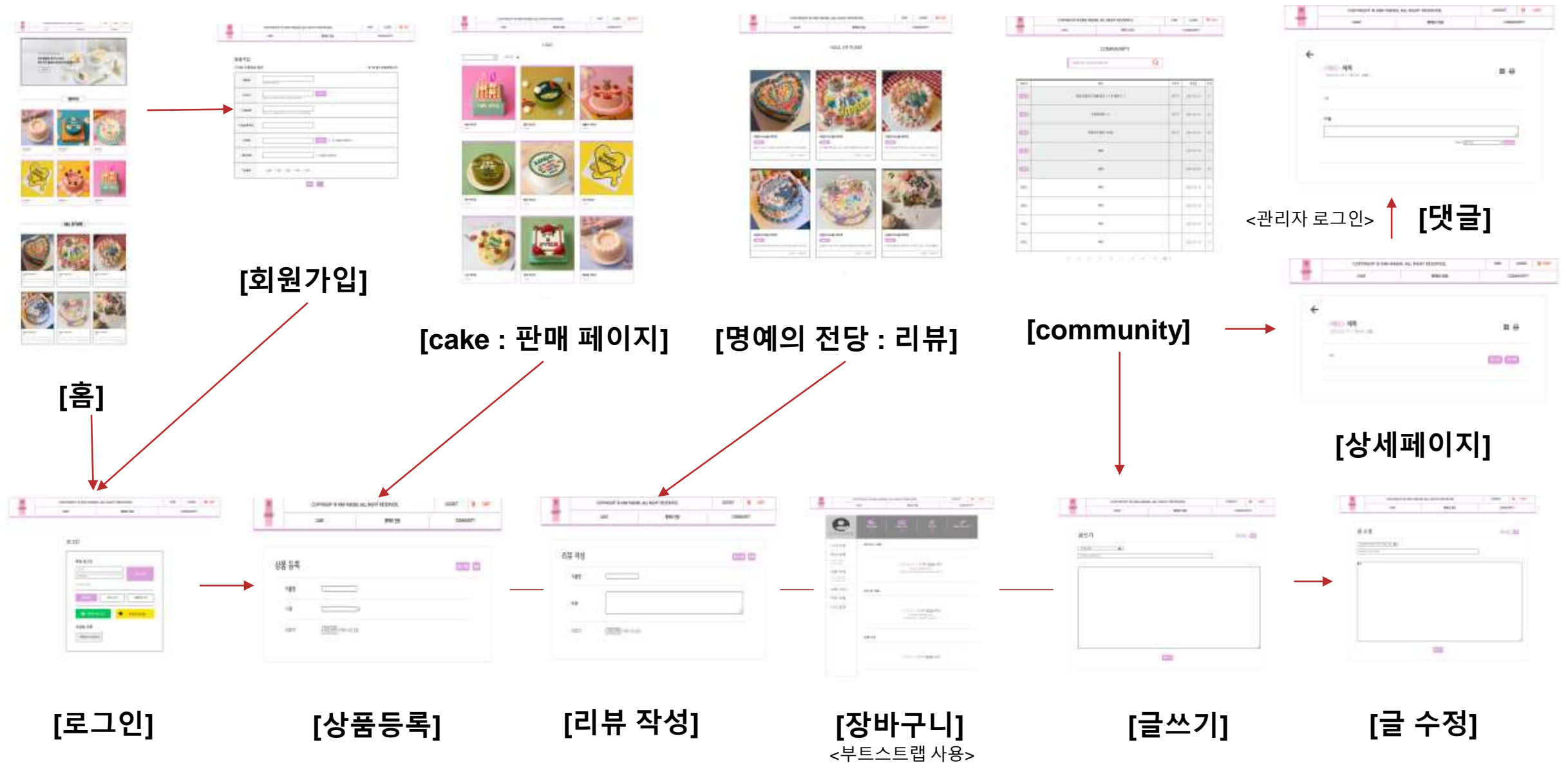
요구사항 정의서

	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	중
	댓글 수정	수정 버튼을 클릭할 시 모달창을 띄운다	필수	하
	댓글 삭제	삭제 버튼을 클릭할 시 해당 댓글을 삭제한다	필수	중
글 수정	권한 확인	글 수정 & 삭제하려는 사용자가 작성자와 일치하는지 확인	필수	중
	글 수정	수정 화면으로 이동한다	필수	상
	글 삭제	글이 삭제되고 목록 페이지로 이동한다	필수	상

Part 4

UI 설계





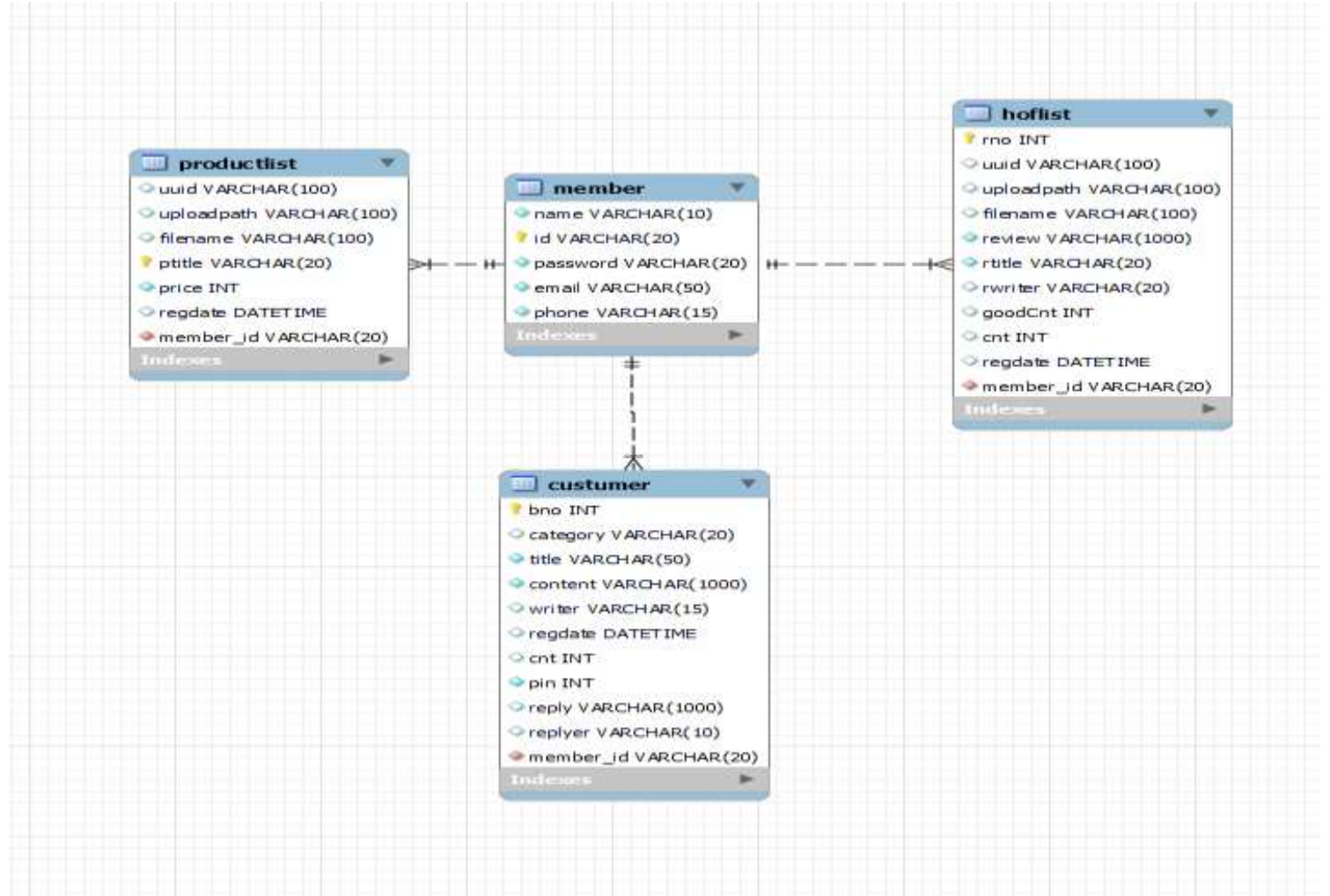
Part 5

DB 설계



ERD

테이블 명	테이블 설명
Product list	제품 테이블
Member	회원 테이블
Hoflist	리뷰 테이블
Customer	게시판 테이블



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

CUSTOMER 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	VARCHAR (1000)			댓글
Replyer	VARCHAR(10)			댓글 작성자

PRODUCTLIST 테이블 명세

Column Name	Datatype	PK	Not Null	설명
UUID	VARCHAR(100)			UUID
UploadPath	VARCHAR(100)			카테고리
Filename	VARCHAR(100)			제목
Ptitle	VARCHAR(20)	○		제품명
Price	Int		○	가격
Regdate	DATETIME			작성 날짜

HOFLIST 테이블 명세

Column Name	Datatype	PK	Not Null	설명
UUID	VARCHAR(100)			UUID
UploadPath	VARCHAR(100)			카테고리
Filename	VARCHAR(100)			파일 이름
Review	VARCHAR(1000)		○	리뷰 내용
Rtitle	VARCHAR(20)		○	리뷰 제목
Regdate	DATETIME			작성 날짜
Rno	INT	○		리뷰 번호
GoodCnt	INT			좋아요
Cnt	INT			조회수

Part 6

주요 기능 상세





<https://github.com/haemikim/PORTFOLIO>

깃허브주소

회원가입

Join.jsp

```
<form action="/join" method="post">
  <table class="join_table">
    <tr>
```

>>

Join.js

```
$(document).ready(function() {
    //회원가입
    $("#join").click(function() {
        //회원가입
        $.ajax({
            url: "/join",
            type: "POST",
            data: $("#joinForm").serialize(),
            success: function(data) {
                //회원가입 성공
                alert("회원가입 성공");
                location.href = "/home";
            },
            error: function(data) {
                //회원가입 실패
                alert("회원가입 실패");
            }
        });
    });
});
```

>>

memberController

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

>>

memberService

```
public interface Mberservice {
    public void insert(MemberDTO mdto);
}
```

↓

memberMapper.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE mapper
PUBLIC "-//mybatis.org/DTD Mapper 3.0//EN"
"http://mybatis.org/dtd/mybatis-3-mapper.dtd">
<mapper namespace="org.cdk.mapper.MemberMapper">
  <insert id="insert">
    insert into member
    values (#{name},#{id},#{password},#{email},#{phone})
  </insert>
</mapper>
```

<<

memberMapper

```
public interface MemberMapper {
    public void insert(MemberDTO mdto);
}
```

<<

memberServiceImpl

```
@Service
public class Mberserviceimpl implements Mberservice{
    @Autowired
    private MemberMapper mmapper;

    public void insert(MemberDTO mdto) {
        mmapper.insert(mdto);
    }
}
```

로그인

login.jsp

```
<form action="/login" method="post">
  <div class="member_box1">
    <div class="id_ps">
```

>>

memberController

```
@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:/home";
    }else{
        return "redirect:/login";
    }
};
```

>>

memberService

```
public interface Mberservice {

    public void insert(MemberDTO mdto);

    public MemberDTO login(MemberDTO mdto);

}
```

V
V

memberMapper.xml

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

<<

memberMapper

```
public interface MemberMapper {

    public void insert(MemberDTO mdto);

    public MemberDTO login(MemberDTO mdto);

}
```

<<

memberServiceImpl

```
@Service
public class Memberserviceimpl implements Mberservice{

    @Autowired
    private MemberMapper mmapper;

    public void insert(MemberDTO mdto) {
        mmapper.insert(mdto);
    }

    public MemberDTO login(MemberDTO mdto) {
        return mmapper.login(mdto);
    }

}
```

로그아웃

Header.jsp

```
<li class="cart">
  <a href="/cart">CART</a>
</li>
```

>>

memberController

```
@GetMapping("logout")
public String logout(HttpSession session){
    session.invalidate();
    return "redirect:/home";
}
```


장바구니

Header.jsp

```
<li class="cart">
  <a href="/cart">CART</a>
</li>
```

>>

memberController

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

>>

Cart.jsp

```
<div test="${login!=null}"><p>${login.id}님 안녕하세요</p></div>
<div>
  <ul>
    <li class="ul1"><a href="#">나의 쇼핑</a></li>
```

주요 상세 기술

글 수정&삭제

Detail.jsp

```
<div class="board_btns">
  <a href="/modify?bno=${detail.bno}">글 수정</a>
  <a href="/remove?bno=${detail.bno}">글 삭제</a>
</div>
```

>>

MainController

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

// <게시판 수정 페이지>에서 수정 버튼을 클릭하면
@PostMapping("modify")
public String modifyPost(BoardDTO board, RedirectAttributes rtr) {
    service.modify(board);
    rtr.addAttribute("bno", board.getBno());
    return "redirect:/detail";
}
```

>>

Modify.jsp

```
<!-- 게시판 수정 폼 -->
<form action="/modify?bno=${bno}" method="post">
    <div class="form-group">
        <input type="text" value="${board.title}" />
    </div>
    <div class="form-group">
        <input type="text" value="${board.content}" />
    </div>
    <div class="form-group">
        <input type="button" value="수정" />
    </div>
</form>
```

Detail.jsp

```
<div class="board_btns">
  <a href="/modify?bno=${detail.bno}">글 수정</a>
  <a href="/remove?bno=${detail.bno}">글 삭제</a>
</div>
```

>>

MainController

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

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

>>

List.jsp

```
<h1 class="st">
  <a href="/list">
    COMMUNITY
  </a>
</h1>
```

댓글쓰기&댓글목록

Join.jsp

```

<!-- 댓글 쓰기 폼 -->
<div class="board_reply">
    <input id="bno" type="hidden" value="${detail.bno}" />
    <div id="replyForm">
        <div id="reply">
            <input type="text" value="" />
            <input type="button" value="댓글 등록" />
        </div>
        <div id="replyList">
            <div id="replyListHeader">
                <div id="replyListHeaderTitle">
                    댓글 목록
                </div>
                <div id="replyListHeaderForm">
                    <input type="text" value="" />
                    <input type="button" value="검색" />
                </div>
            </div>
            <div id="replyListContent">
                <table border="1">
                    <thead>
                        <tr>
                            <th>번호</th>
                            <th>내용</th>
                            <th>작성자</th>
                            <th>작성일</th>
                        </tr>
                    </thead>
                    <tbody>
                        <tr>
                            <td>1</td>
                            <td>댓글 1</td>
                            <td>admin</td>
                            <td>2023-01-01</td>
                        </tr>
                    </tbody>
                </table>
            </div>
        </div>
    </div>
</div>

```

>>

Detail.js

```

var BoardService = function() {
    // 댓글 쓰기 폼
    function update(reply, callback) {
        console.log(reply);
        $.ajax({
            url: "/board",
            type: "put",
            data: JSON.stringify(reply),
            contentType: "application/json",
            success: function(result) {
                callback(result);
            },
            error: function() {}
        });
    }

    // 댓글 목록
    function getList(param, callback) {
        var bno = param.bno;
        $.getJSON("/board/" + bno + ".json", function(data) {
            callback(data);
        });
    }
}

```

>>

MainController

```

@RequestMapping(value="/board", produces="application/json", consumes="application/json")
public void update(@RequestBody BoardDTO board) throws IOException {
    // 댓글 쓰기
    boardService.update(board);
    // 댓글 목록
    boardService.getList(board.getBno());
}

@RequestMapping(value="/board/{bno}", produces="application/json", consumes="application/json")
public void getList(@PathVariable("bno") int bno) throws IOException {
    // 댓글 목록
    boardService.getList(bno);
}

```

v

Detail.js

```

// 댓글 쓰기 폼
$(function() {
    // 댓글 쓰기 폼
    $("#replyForm").on("click", function() {
        // 댓글 쓰기
        var reply = $("#reply").val();
        var replyer = $("#replyer").val();
        // 댓글 등록
        boardService.update(reply, replyer, bno, bno, function(result) {
            alert("댓글 등록 성공");
            console.log(result);
            showList();
        });
    });

    // 댓글 목록
    $(function() {
        // 댓글 목록
        $("#replyListHeaderForm").on("click", function() {
            // 댓글 목록
            var bno = $("#bno").val();
            boardService.getList(bno);
        });
    });
});

```

<<

BoardMapper &
BoardMapper.xml

```

// 댓글쓰기 설계
// 값을 리턴받는다
public int Rwrite(BoardDTO board);
// 댓글 목록 리스트 설계
public BoardDTO Rlist(int bno);

```

<<

```

<update id="Rwrite">
    update customer
    set reply = #{reply},
    replyer = #{replyer}
    where bno = #{bno}
</update>

<select id="Rlist" resultType="org.mybatis.domain.BoardDTO">
    select * from customer where bno=#{bno}
</select>

```

memberService &
memberServiceImpl

```

// 게시판 댓글쓰기 설계문장을 구현
public int Rwrite(BoardDTO board);
// 댓글 목록 리스트 설계
public BoardDTO Rlist(int bno);

// 게시판 댓글쓰기 설계문장을 구현
public int Rwrite(BoardDTO board) {
    // insert 문장시 ReplyMapper.xml로 부터
    // insert 문장시 ReplyMapper.xml로 부터
    // 값을 리턴받는다
    return mapper.Rwrite(board);
}

// 댓글 목록 리스트 설계문장을 구현
public BoardDTO Rlist(int bno) {
    return mapper.Rlist(bno);
}

```

댓글 수정&삭제

Join.jsp

```

<!-- 댓글 등록 -->
<div class="form" style="border: 1px solid #ccc; padding: 10px; margin-bottom: 10px;">
    <input id="bno" type="text" value="${detail.bno}" />
    <input id="reply" type="text" value="" />
    <input type="button" value="댓글 등록" />

```

```

<input id="session" type="hidden" value="${login.id}" />
<div id="relist">

```

>>

Detail.js

```

// 댓글 등록
function registerReply() {
    $.ajax({
        url: "${contextPath}/board/reply.do",
        type: "POST",
        data: {
            "bno": $("#bno").val(),
            "reply": $("#reply").val()
        },
        success: function(data) {
            alert("댓글 등록 성공");
        },
        error: function() {
            alert("댓글 등록 실패");
        }
    });
}

```

>>

MainController

```

// 댓글 등록
@RequestMapping(value="/board/reply.do", method=RequestMethod.POST)
public void registerReply(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
    // 댓글 등록
    BoardDTO board = new BoardDTO();
    board.setBno(Integer.parseInt(request.getParameter("bno")));
    board.setReply(request.getParameter("reply"));
    memberService.registerReply(board);
    response.sendRedirect(request.getContextPath() + "/board/detail.do");
}

```

↓
↓

Detail.js

```

// 댓글 등록
function registerReply() {
    $.ajax({
        url: "${contextPath}/board/reply.do",
        type: "POST",
        data: {
            "bno": $("#bno").val(),
            "reply": $("#reply").val()
        },
        success: function(data) {
            alert("댓글 등록 성공");
        },
        error: function() {
            alert("댓글 등록 실패");
        }
    });
}

```

```

// 댓글 삭제
function removeReply() {
    $.ajax({
        url: "${contextPath}/board/reply.do",
        type: "DELETE",
        data: {
            "bno": $("#bno").val()
        },
        success: function(data) {
            alert("댓글 삭제 성공");
        },
        error: function() {
            alert("댓글 삭제 실패");
        }
    });
}

```

<<

BoardMapper & BoardMapper.xml

```

<update id="Rupdate">
    update customer
    set reply = #{reply}
    where bno = #{bno}
</update>

<update id="Rremove">
    update customer
    set reply = NULL
    where bno = #{bno}
</update>

```

```

// 댓글 update 설정
public int Rupdate(BoardDTO board);
// 댓글 삭제 설정
public int Rremove(BoardDTO board);

```

<<

memberService & memberServiceImpl

```

// 댓글 update 설정
public int Rupdate(BoardDTO board);
// 댓글 삭제 설정
public int Rremove(BoardDTO board);

```

```

// 댓글 update 설정
public int Rupdate(BoardDTO board) {
    return mapper.Rupdate(board);
}
// 댓글 삭제 설정
public int Rremove(BoardDTO board) {
    return mapper.Rremove(board);
}

```


페이징&검색

jsp

```

<form id="actionForm" action="/list" method="get">
  <fieldset class="hd_field">
    <legend>검색/페이지</legend>
    <input type="text" name="keyword" value="${pageMaker.cri.keyword}" id="search_board" class="form-control">
    <button type="submit" class="btn btn-search">
      <span class="glyphicon glyphicon-search">
    </button>
  </fieldset>
  <input type="hidden" name="pageNum" value="${pageMaker.cri.pageNum}>
  <input type="hidden" name="amount" value="${pageMaker.cri.amount}>
</form>

```

>>

Controller

```

// <게시판 목록 페이지>로 리턴
@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));
}

```

>>

Service

```

// 게시판 페이징에 쓰일 페이징 전체
public int getTotalCount(Criteria cri);
// 게시판 글쓰기 선택한 것을 리턴

```

↓
↓

Mapper.xml

```

<select id="list" resultType="org.springframework.data.domain.PageDTO">
  select *
  from (
    select @pageNum=@pageNum+1 as pageNum, c.*
    from (select @pageNum:=0) as tmp,
    customer as c
    <if test="@keyword" != null">
      where title like concat('%',#{keyword},'%')
      or content like concat('%',#{keyword},'%')
      or writer like concat('%',#{keyword},'%')
    </if>
    order by pin desc, regdate desc
  ) as boardList
  <if test="@pageNum" != null">
    where pageNum >= #{@pageNum} and pageNum <= #{@pageNum}+#{@amount}
  </if>
</select>

```

<<

Mapper

```

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

```

<<

ServiceImpl

```

// 게시판 페이징에 쓰일 데이터 건수
public int getTotalCount(Criteria cri) {
    return bmapper.getTotalCount(cri);
}

```

Cake제품페이지리스트

ProductList.jsp

>>

ProductController

```
// 4. 为每个请求返回一个模型对象
@GetMapping("productlist")
public void productlist(Criteria cri, Model model) {
    System.out.println("productlist="+cri);
    model.addAttribute("productlist", Pservice.productlist(cri));
    int total=Pservice.getTotalCount(cri);
    model.addAttribute("pageMaker", new pageDTO(cri,total));
}
```

>>

ProductService

```
// 제품 목록 리스트 설계
public ArrayList<ProductDTO> productList(Criteria cri);
```

VV

ProductMapper.xml

[illegible]

<<

ProductMapper

```
// 제품 목록 리스드 없게
public ArrayList<ProductDTO> productList(Criteria cri);
```

<<

ProductServiceimpl

```
// 제품 목록 리스츠 검색
public ArrayList<ProductDTO> productList(Criteria cri){
    return pmapper.productList(cri);
}
```

명예의전당리스트페이지(조회수)

HallOfFame.jsp

```
<!-- 로그인 후 그에게 보도록 우선순위에 맞는 명예의전당 만들기 -->
<form id="actionForm" action="/hallOfFame" method="get">
```

>>

HOFController

```
// <제품 목록 페이지>로 리다이렉트
@GetMapping("hallOfFame")
public void hallOfFame(Criteria cri, Model model) {
    System.out.println("hallOfFame=" + cri);
    model.addAttribute("hallOfFame", hservice.hallOfFame(cri));
    model.addAttribute("pageMaker");
}
```

>>

HOFService

```
// 제품 목록 리스트 설계
public ArrayList<HOFDTO> hallOfFame(Criteria cri);
```

↓

HOFMapper.xml

```
<select id="hallOfFame" resultType="org.saebyeol.domain.hof.HOFDTO">
    select uuid,
           replace(uploadpath, '\\', '/') uploadpath,
           filename,
           rtile,
           cnt,
           review,
           regdate
    from(
        select @rownum:=@rownum+1 as rownum, ht.*
        from (select @rownum:=0) as tmp,
             HOFList as ht
        order by goodCnt, regdate desc
    ) as HOFList
    <![CDATA[
        where rownum > #{(pageNum-1)*@amount}
        and rownum <= #{@pageNum*@amount}
    ]]>
</select>

<update id="CNTupdate">
    update HOFList
    set cnt = cnt+1
    where no = #{no}
</update>
```

<<

HOFMapper

```
// 제품 목록 리스트 설계
public ArrayList<HOFDTO> hallOfFame(Criteria cri);
```

```
// 게시판 목록리스트에서 제목을 클릭했을때 조회수 +1
public void CNTupdate(HOFDTO board);
```

<<

HOFServiceimpl

```
// 제품 목록 리스트 설계
public ArrayList<HOFDTO> hallOfFame(Criteria cri){
    return hmapper.hallOfFame(cri);
}
```

100

jsp

[illegible]

>>

js

```

1 // @flow
2 import {type: 'file'} from '@types/mime';
3 function() {
4   Alert('XSS XX-7');
5
6   const formData = new FormData();
7   const inputFile = document.querySelector('input[type="file"]');
8   const files = inputFile.files;
9
10  for(let i=0; i<files.length; i++) {
11    // @flow
12    if(SecurityUtil.isSafeFile(files[i].size, files[i].name)) {
13      inputFile.value = files[i].name;
14      return false;
15    }
16    formData.append('upload[]', files[i]);
17  }
18  const {
19    url: 'upload',
20    type: 'post',
21    data: formData,
22    processData: false,
23    contentType: false,
24    success: function(data) {
25      showModal('success');
26    }
27  }
28 }

```

V
V

jsp

```
img_src = "%s(%s(%s/%s))%s" % (img_src, '%(L1LT_aplmodPath)s', '%(L1LT_uid)s', '%(L1LT_filename)s', ".d") % d1
```

< <

Controller

```

1  #!/usr/bin/perl
2
3  use strict;
4  use warnings;
5
6  my $url = "http://www.example.com";
7  my $user_agent = "Mozilla/5.0 (Windows NT 6.0; rv:2.0) Gecko/20100101 Firefox/4.0";
8
9  my $response = LWP::UserAgent->new(
10     agent => $user_agent,
11     proxy => {}
12 )->get($url);
13
14 my $html = $response->content;
15
16 # Parse the HTML
17 my $doc = HTML::Dom->new($html);
18
19 # Find all links
20 my @links = $doc->getElementsByTagName("a");
21
22 foreach my $link (@links) {
23     my $href = $link->getAttribute("href");
24     my $text = $link->getInnerText;
25
26     # Print the link
27     print "$text ($href)\n";
28 }
29
30 # Close the document
31 $doc->close();

```

Cake 제품 추가 페이지 & 리뷰 추가 페이지

ProductList.jsp

```
<a href="/addProduct">
```

>>

ProductController

```
// <제품 등록 페이지>로 이동
@GetMapping("addProduct")
public void addProduct() {
    |
}
```

>>

AddProduct.jsp

```
<form role="form" class="ap_form" action="/addProduct" method="post">
```

HallOfFame.jsp

```
<a href="/addHOF">
```

>>

HOFController

```
// <제품 등록 페이지>로 이동
@GetMapping("addHOF")
public void addHOF() {
    |
}
```

>>

AddHOF.jsp

```
<form role="form" class="ap_form" action="/addHOF" method="post">
```



감사합니다