**2조 스프링 프로젝트**

**더존비즈온 3기**

**2조 박승민 안예지 홍건호 류호진**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | | | | |
| **[ 목 차 ]** | | | | |
| **제1절 프로젝트 개요** | | |  |  |
|  | 1.1 개발배경 및 목적 | | ‥‥‥‥‥‥‥‥‥‥‥‥‥‥‥‥‥‥‥‥‥ | 3 |
|  | 1.2 추진체계 및 일정 | | ‥‥‥‥‥‥‥‥‥‥‥‥‥‥‥‥‥‥‥‥‥ | 3 |
|  | 1.3 부분별 수행업무 및 담당자 | | ‥‥‥‥‥‥‥‥‥‥‥‥‥‥‥‥‥‥‥‥‥ | 4 |
|  | 1.4 프로젝트 개발 범위 | | ‥‥‥‥‥‥‥‥‥‥‥‥‥‥‥‥‥‥‥‥‥ | 4 |
| **제2절 시스템 구축내용** | | |  |  |
|  | 2.1 시스템 구성도  2.2 시스템 흐름도 | | ‥‥‥‥‥‥‥‥‥‥‥‥‥‥‥‥‥‥‥‥‥  ‥‥‥‥‥‥‥‥‥‥‥‥‥‥‥‥‥‥‥‥‥ | 5 |
| **제3절 시연** | | |  | 6 |
|  | | 3.1 테스트 및 검증 | ‥‥‥‥‥‥‥‥‥‥‥‥‥‥‥‥‥‥‥‥‥ | 6 |
|  | | 3.2 Git & Notion | ‥‥‥‥‥‥‥‥‥‥‥‥‥‥‥‥‥‥‥‥‥ | 7 |
|  | |  |  |  |
| **제4절 개발내용** | | |  |  |
|  | 4.1 DB 테이블 설정 | | ‥‥‥‥‥‥‥‥‥‥‥‥‥‥‥‥‥‥‥‥‥ | 8 |
|  | 4.2 mvc 패턴 설계 방식 (해설) | | ‥‥‥‥‥‥‥‥‥‥‥‥‥‥‥‥‥‥‥‥‥ | 12 |
|  |  | |  |  |
| **제5절 결론 및 개선방안** | | |  |  |
|  | 5.1 결론 및 기대효과 | | ‥‥‥‥‥‥‥‥‥‥‥‥‥‥‥‥‥‥‥‥‥ | 15 |
|  | 5.2 문제점 및 개선 사항 | | ‥‥‥‥‥‥‥‥‥‥‥‥‥‥‥‥‥‥‥‥‥ | 15 |
|  |  | |  |  |

제1절 프로젝트 개요

1.1 개발 배경 및 목적

|  |  |
| --- | --- |
| 기획배경 | 헬스장에서 운동을 하기 전 “필요한 것 들을 모두 해결해줬으면 좋겠다 “ 는 생각에서 접근 |
| 기획목적 | 기존의 헬스장이란 틀에서 벗어나 운동뿐만 아닌 영양제 구독 서비스 , 맞춤형 식단과 음료, 용품 등을 고객에게 제공 |
| 기대효과 | 운동하는 고객들의 맞춤형 건강식 및 영양제 구독 서비스를 제공하며, 여러 협력 업체들과의 협업 및 가맹점 추가로 인한 기대수익 증가 |
| 기능요약 | 영양제 구독 서비스 제공 / 물품 구매 기능 식단 구매 기능/로그인을 활용한 개인 맞춤 추천 서비스/ 장바구니를 통한 구매/구매 목록 확인 기능 /Q &A (1대1 문의)/ 관리자 (물품 ,식단, 영양제, 구매 ,문의) 수정,추가 그리고 삭제 |

|  |
| --- |
| 개발 Tool |
| Spring Tool Suite 3 |
| Oracle SQL developer |

<표1.1> 사용 도구

1.2 추진체계 및 일정

팀장 : 류호진

팀원 : 박승민, 안예지, 홍건호

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | 10  31 | 11  01 | 11  02 | 11  03 | 11  04 | 11  05 | 11  06 | 11  07 | 11  08 | 11  09 | 11  10 | 11  11 | 11  12 | 11  13 | 11  14 |
| 설  계 | 주제선정 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| DB 테이블 작성 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 프  로  그  램  구  현 | 기능 세분화 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 우선순위 설정 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 1차 기능구현 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 1차 머지 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 2차 기능구현 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 2차 머지 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 추가 기능구현 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| css작업 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 테  스  트 | 문서 정리 및 PPT작성 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

<표1.2> 프로젝트 일정표

1.3 부분 별 수행 업무 수행 및 담당자

|  |  |  |  |
| --- | --- | --- | --- |
| **분 야** | **담당 업무** | **사용 언어** | **담 당 자** |
| 프로젝트 총괄 | 로그인 기능, 회원가입, 설문(조사, 결과 ,조회),  인바디(입력, 결과 ,조회),  결제 (확인) | CSS, JAVA,JSP,JAVASCRIPT  Oracle SQL | 류호진 |
| 서브 기능 총괄 | 관리자 페이지 qna조회,문의 페이지 faq, 1대1문의, 상품 검색,부분 검색, css작업, 설문(결과 연계 추천), 인바디(결과 연계 추천) | CSS, JAVA,JSP,JAVASCRIPT  Oracle SQL | 박승민 |
| 관라자페이지 총괄 | 관라자페이지, 회원조회, 상품관리, 상품등록, 주문조회, css작업, 메인페이지 연결 설정, 식품 장바구니 연결, 용품 장바구니 연결 | CSS, JAVA,JSP,JAVASCRIPT  Oracle SQL | 안예지 |
| 메인 프레임 총괄 | 영양제 구독, 영양제 장바구니 연결, 장바구니 구현, 결제 구현, 본인인증 구현, css작업, 메인틀 구축, js처리 | CSS, JAVA,JSP,JAVASCRIPT  jQuery,Oracle SQL | 홍건호 |

<표1.3> 프로젝트 업무수행 및 담당자

1.4 프로젝트 개발 범위

로그인 및 회원가입은 메인페이지 우측 상단 버튼을 통해 할 수 있고 회원가입시 본인인증을 해야 회원가입을 진행할 수 있다. 회원가입시 정규표현식을 만족 해야 가입이 가능하다.

로그인 시 아이디가 admin이면 관리자 페이지로 이동

관리자는 1. 회원조회 2. 상품관리(update, delete) 3.상품등록 4. 주문조회 5. Q&A 확인

들을 할 수 있다.

사이트는 소개, 문의, 마이페이지, 상품, Q&A로 이루어져 있다.

소개는 텍스트와 이미지 동영상을 이용해 헬스장과 be natural(홈페이지 이름)의 소개가 적혀 있다.

문의는 자주 묻는 질문들과 그 답변들이 정리되어 있는 FAQ 와 1대1 문의 로 나뉘어 지는데 1대 1 문의의 답변은 관리자 페이지를 통해 할 수 있다.

마이페이지는 로그인을 한 뒤 메인페이지에서 로그인 버튼을 다시 누르면 갈 수 있는데 마이페이지에서는 본인의 아이디와 이름 성별을 확인할 수 있고 본인의 설문조사리스트, 인바디리스트, 구매리스트로 이동과 로그아웃 기능을 활용할 수 있다.

설문조사리스트,인바디리스트에서는 설문조사,인바디 값 입력을 통해 결과에 맞는 상품들을 추천 받을 수 있다.

상품들은 3개로 나누어져 있다. 헬스용품, 헬스식품, 헬스영양제로 이루어져 있다. 각 제품들은 카테고리별로 나누어져 있고 카테고리 별로 제품들을 확인할 수 있다. 각 제품에 마우스커서 가 올라가면 css로 마우스가 올라가면 색을 바꿔 표시가 되고 클릭 시 장바구니 담기가 있는 상세보기 페이지로 넘어간다.

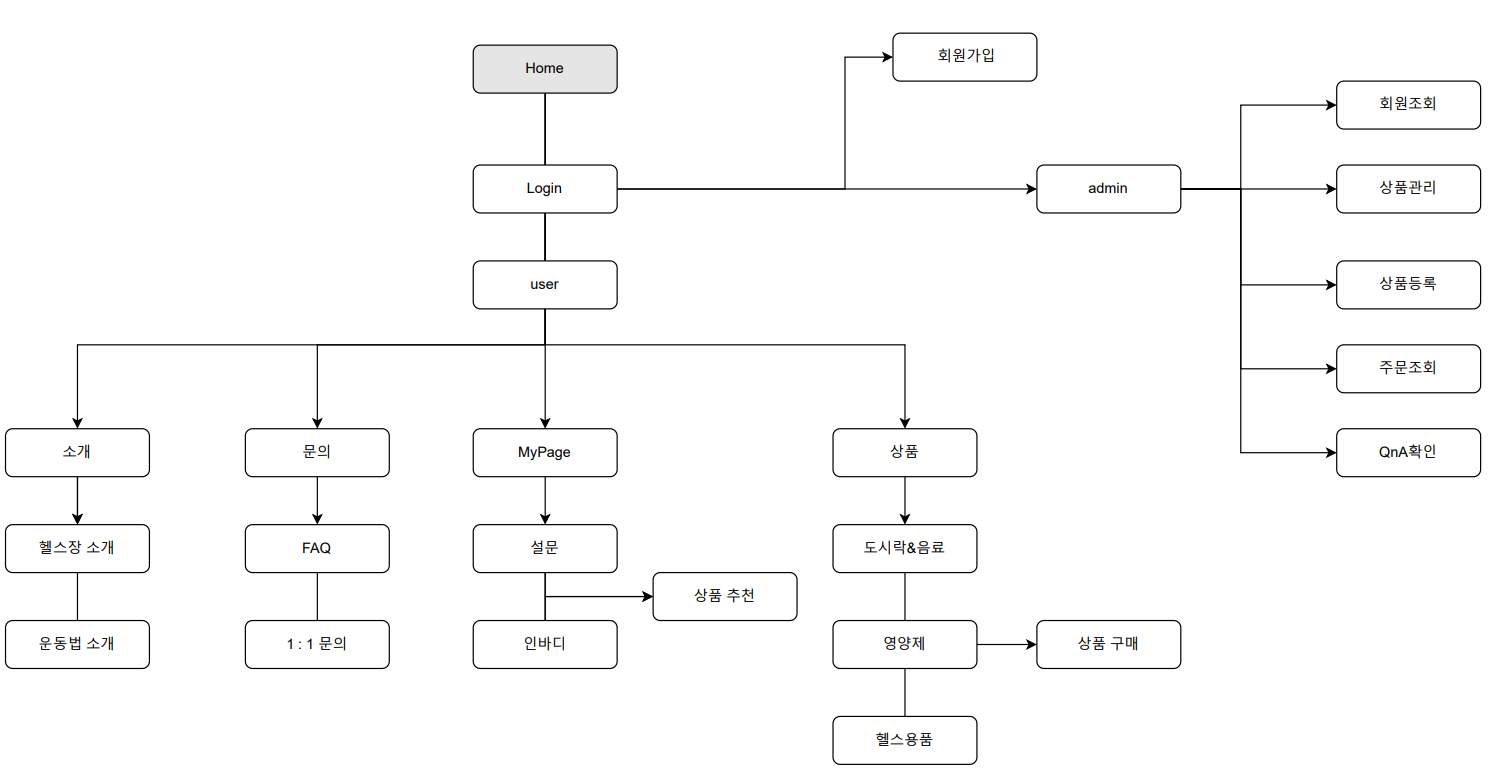
제품 상세보기 페이지에서는 수량을 결정한 뒤 장바구니에 담을 수 있다. 이때, 음수는 담을 수 없게 처리를 했다.

용품과 식품은 구독을 서비스를 제공하지 않지만 영양제에서는 구독 서비스를 제공한다. 영양제는 구독과 즉시구매로 나뉘어 지고 한 통구매와 1회분 구매로 나뉘어 진다. 한 통 구매는 1병이 제공하는 회분을 한꺼번에 구매 할 수 있다.

장바구니 페이지에서는 회원이 담아둔 장바구니를 저장하고 물품을 체크박스로 삭제 할 수 있다. 구매는 신용카드 구매와 카카오 페이 구매가 있는데 구매에 실패하면 다시 장바구니로 오게 되고 성공하면 구매 확인 페이지로 이동하게 된다. 구매확인 페이지에서는 마이페이지 구매리스트 보기로 이동할 수 있다.

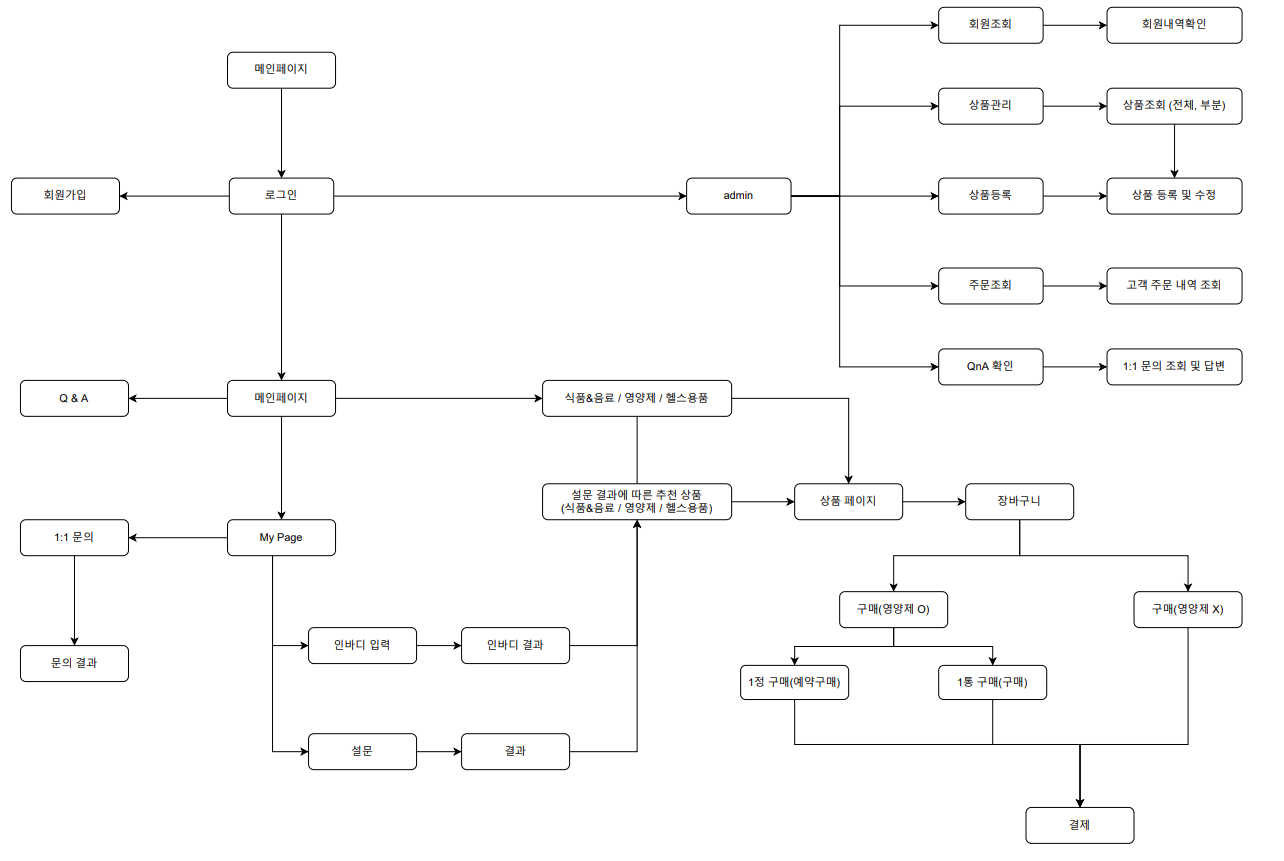
제2절 시스템 구축내용

2.1 시스템 구성도



<그림2.1 구성도>

2.2 시스템 흐름도



<그림2.2 시스템 흐름도>

제3절 시연

3.1 테스트 및 검증

<https://github.com/rhj1216-hochan06/douzone_th3_2team> 참조

<https://www.notion.so/2-f233f21483664ca7ae3b1afbdcbb8d39> 세부사항 참조

3.2 Git & Notion

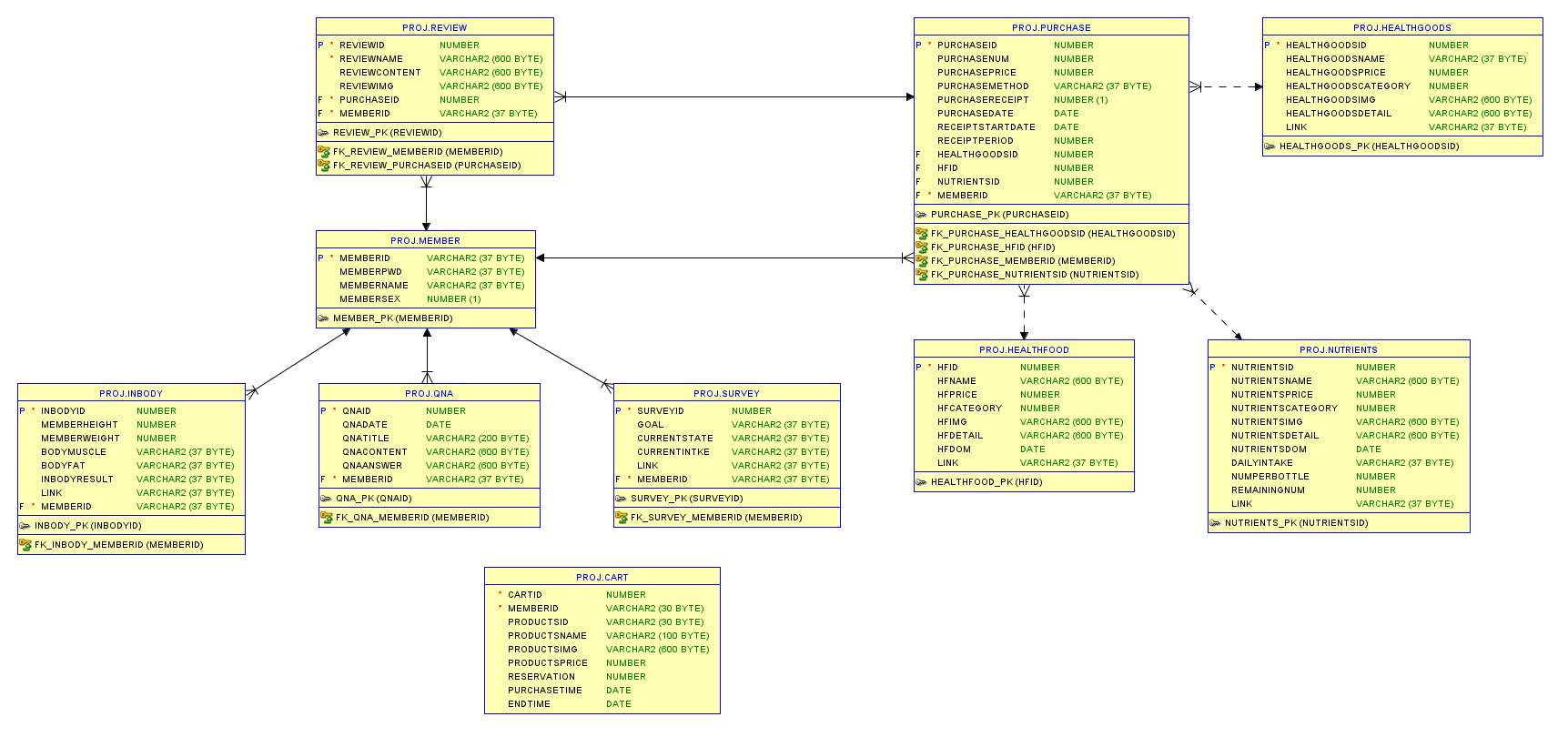


<그림 3.2 Git 소스트리>

<그림 3.2 Git 소스트리>

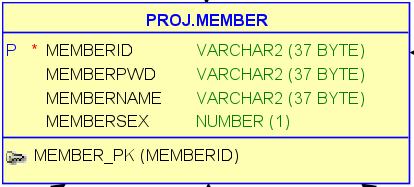


제4절 개발내용

4.1 DB 테이블 설정

<그림4.1 DB 전체 테이블>

1. **Member**



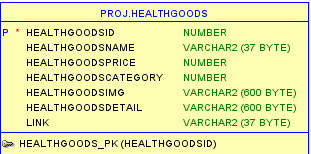
MemberID varchar2(37) primary key, --회원 id max 20 자바에서 정규화 처리

MemberPwd varchar2(37), --회원 pw 자바에서 정규화 처리

MemberName varchar2(37), --회원 이름 자바에서 정규화 처리

MemberSex number(1) --회원 성별 (1 or 0)

1. **healthgoods**



HealthgoodsID number primary key, --헬스용품 id

HealthgoodsName varchar2(37), --헬스용품 이름

HealthgoodsPrice number, --헬스용품 가격

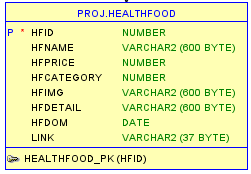
HealthgoodsCategory number, --헬스용품 카테고리

HealthgoodsIMG varchar2(600), --헬스용품 이미지

HealthgoodsDetail varchar2(600), --헬스용품 상세설명

Link varchar2(37) --설문조사 링크

1. **healthfoods**



HFID number primary key, --식품 id

HFName varchar2(600), --식품 이름

HFPrice number, --식품 가격

HFCategory number, --식품 카테고리

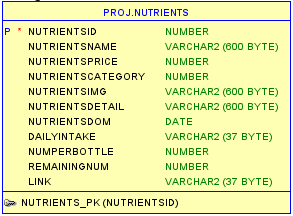
HFIMG varchar2(600), --식품 이미지

HFDetail varchar2(600), --식품 상세설명

HFDoM Date, --식품 제조날짜

Link varchar2(37) --인바디,설문 연결값

1. **nutrients**



NutrientsID number primary key, --영양제 id

NutrientsName varchar2(600), --영양제 이름

NutrientsPrice number, --영양제 가격

NutrientsCategory number, --영양제 카테고리

NutrientsIMG varchar2(600), --영양제 이미지

NutrientsDetail varchar2(600), --영양제 상세설명

NutrientsDoM Date, --영양제 제조날짜

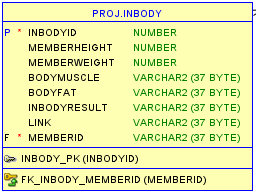
DailyIntake varchar2(37), --1일 섭취 갯수

NumPerBottle number, --1통 당 갯수

RemainingNum number, --현재 잔여 갯수

Link varchar2(37) --인바디,설문 연결값

1. **inbody**



InBodyId number primary key, --인바디 id

MemberHeight number, --키

MemberWeight number, --몸무게

BodyMuscle varchar2(37), --근육량

BodyFat varchar2(37), --체지방량

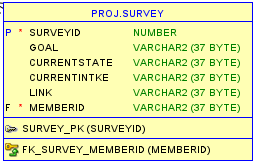
InbodyResult varchar2(37), --상태 결과 (근골격 \* 체지방 / 2 = 점수)

Link varchar2(37), --결과 제품 추천용

MemberID varchar2(37) NOT NULL, --회원id (외래키)

CONSTRAINT fk\_inbody\_MemberID FOREIGN KEY(MemberID) references Member (MemberID)

1. **survey**



SurveyId number primary key, --설문 id

Goal varchar2(37), --목표 (증량/감량)

CurrentState varchar2(37), --현재 상태 (비만/정상/왜소)

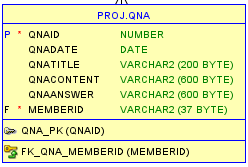
CurrentIntke varchar2(37), --현재 섭취 상태 (과다/정상/부족)

Link varchar2(37), --결과 제품 추천용

MemberID varchar2(37) NOT NULL, --회원id (외래키)

CONSTRAINT fk\_Survey\_MemberID FOREIGN KEY(MemberID) references Member (MemberID)

1. **qna**



QnAId number primary key, --QnA id

QnADate date, --QnA 날짜

QnATitle varchar2(200), --QnA 제목

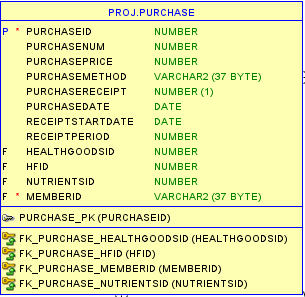
QnAContent varchar2(600), --QnA 내용

QnAAnswer varchar2(600), --QnA 답변

MemberID varchar2(37) NOT NULL, --회원 id (외래키)

CONSTRAINT fk\_QnA\_MemberID FOREIGN KEY(MemberID) references Member (MemberID)

1. **purchase**



PurchaseId number primary key, --구매 id

PurchaseNum number, --구매 갯수

Purchaseprice number, --구매 가격

PurchaseMethod varchar2(37), --결제 방식 (카드/계좌이체)??

PurchaseReceipt number(1), --수령 방식 (즉시/예약)

PurchaseDate date, --구매(결제) 날짜

ReceiptStartDate date, --수령(픽업) 시작일 (ex:2022-10-23)

ReceiptPeriod number, --수령(픽업) 기간 (ex: 7) -> 2022-10-23부터 시작하여 7일간 픽업할 수 있도록 데이터

HealthgoodsID number, --헬스용품 id (외래키)

HFID number, --식품 id (외래키)

NutrientsID number, --영양제 id (외래키)

MemberID varchar2(37) NOT NULL, --회원id (외래키)

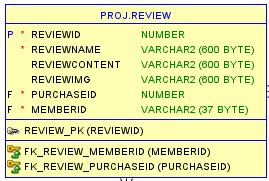
CONSTRAINT fk\_Purchase\_HealthgoodsID FOREIGN KEY(HealthgoodsID) references Healthgoods(HealthgoodsID),

CONSTRAINT fk\_Purchase\_HFID FOREIGN KEY(HFID) references HealthFood (HFID),

CONSTRAINT fk\_Purchase\_NutrientsID FOREIGN KEY(NutrientsID) references Nutrients (NutrientsID),

CONSTRAINT fk\_Purchase\_MemberID FOREIGN KEY(MemberID) references Member (MemberID)

1. **review**



ReviewId number primary key, --구매후기 id

ReviewName varchar2(600) NOT NULL, --구매후기 제목

ReviewContent varchar2(600), --구매후기 내용

ReviewIMG varchar2(600), --구매후기 이미지

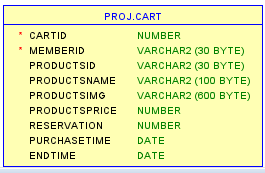
PurchaseID number NOT NULL, --구매 id (외래키)

MemberID varchar2(37) NOT NULL, --회원 id (외래키)

CONSTRAINT fk\_Review\_MemberID FOREIGN KEY(MemberID) references Member (MemberID),

CONSTRAINT fk\_Review\_PurchaseID FOREIGN KEY(PurchaseID) references Purchase (PurchaseID)

1. **cart**



memberId varchar2(30) not null, -- 고객 아이디

productsId varchar2(30), -- 물품 아이디

productsName varchar2(100), -- 물품 이름

productsIMG varchar2(100), -- 물품 사진 주소

productsPrice number, -- 물품 가격

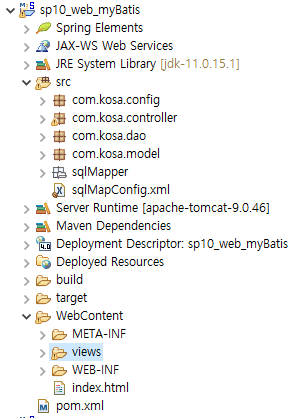
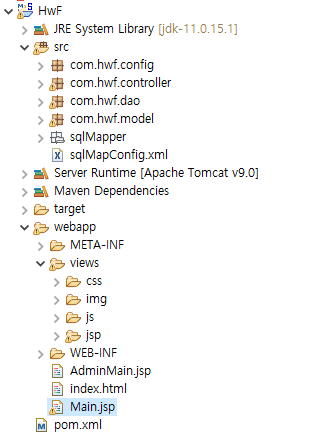
reservation number, -- 즉시, 예약 선택

purchasetime date, -- 구매할 날짜

Endtime date -- 수령이 끝나는 날짜

4.2 자바 클래스 및 메소드 설정

화면 UI 구성을 생각하고 UI에 맞춰서 메인틀을 구현하였습니다. 프로젝트를 진행을 하면서 메인틀을 가지고 시작하였습니다. 먼저 대부분의 페이지에서 사용될 공통적인 부분들을 생각하면서 구현 하였고 프로젝트의 틀은 수업시간에 배운 sp10\_web\_myBatis를 참조하였습니다.



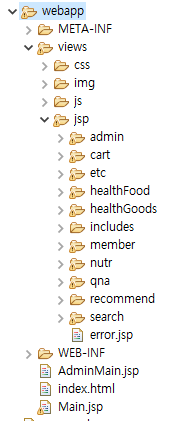
sp10\_web\_myBatis는 크게 src와 WebContent로 나뉘어지고

1. src
   1. config
   2. controller
   3. dao
   4. model
   5. sqlmapper
2. WebContent
   1. Mmeta-inf
   2. Views
   3. Web-inf

<그림4.1 sp10\_web\_myBatis와 이번 프로젝트의 구성>

로 이루어집니다.

이번 프로젝트의 형식도 sp10\_web\_myBatis와 비슷합니다

프로젝트 작업순서는 아래와 같습니다.

프로젝트를 진행함에 있어서 먼저 설계해둔 기능별로 흐름도와 구성도를 보면서 먼저 화면(jsp 파일)을 제작 합니다.

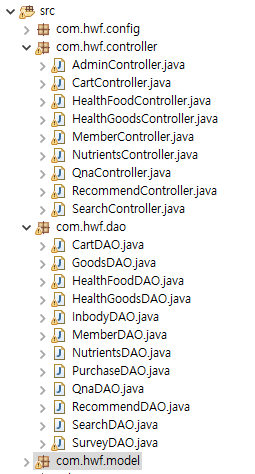
그 이후 화면을 호출할 때 사용할 컨트롤러를 생성합니다. 메인틀을 제작 할 때 만든 config들을 사용하여 필요한 데이터들을 가져오는 mapper를 제작하고, mapper에서 불러오는 데이터 값들을 dto타입으로 가져 오기 위해 dto를 제작합니다.

Dto와 mapper를 연결하는 dao를 제작하고 컨트롤러에서 dao를 호출하여 기능을 구현하고 컨트롤러에서 이동할 페이지를 호출합니다.

jsp에서 컨트롤러를 호출할 때는 웹 Servlet 방식으로 온 클릭 메소드를 통해 컨트롤러를 호출 합니다.

Ex) <a onclick = " location.href = 'Member?cmd=login' ">//로그인으로 이동

<그림4.1 이번 프로젝트의 페이지의 구성>

4.3 주요 로직 설명

로그인 및 회원가입은 메인페이지 우측 상단 버튼을 통해 할 수 있고 회원가입시 본인인증을 해야 회원가입을 진행할 수 있다. 회원가입시 정규표현식을 만족해야 가입이 가능하다.

로그인 정보는 member.dao에서 public MemberDTO checkMember (MemberDTO dto)함수를 활용하여 아이디와 비번을 입력하여 호출된 아이디와 비번이 일치하는지 확인하여 일치하면 메인으로

request.getRequestDispatcher("/Main.jsp").forward(request, response);

아니면 로그인 화면으로 다시 보내고 비밀번호가 틀렸다고 표시해준다.

로그인 프로세스를 진행시 세션을 활용하여

HttpSession session;

session = request.getSession();

session.setMaxInactiveInterval(20 \* 60); // 세션 유지시간 설정(초단위) 20분

로그인 정보를 비밀번호를 제외하고 저장하여 다른 페이지에서 회원 아이디와 이름이 필요한 경우 호출하여 사용한다.

로그인 시 아이디가 admin이면 관리자 페이지로 이동

관리자는 1. 회원조회 2. 상품관리(update, delete) 3.상품등록 4. 주문조회 5. Q&A 확인

들을 할 수 있다.

여기서 2번 상품관리를 본다면 먼저 카테고리를 선택하면 컨트롤러로 이동한다. 컨트롤러에서

GoodsDAO dao = **new** GoodsDAO();

List<NutrientsDTO> AdminNutrientsSelect = dao.AdminNutrientsSelect();

위와 같은 코드를 통하여 dto 타입의 리스트를

request.setAttribute("AdminNutrientsSelect", AdminNutrientsSelect);

리퀘스트의 어트리뷰트로 넘겨준다.

그후 jsp로 이동한 후 어트리뷰트를 받아주면 리스트가 출력된다.

받아주는 법은 아래와 같다

<c:forEach var="AdminNutrientsSelect" items="${ AdminNutrientsSelect }">

<tr class="row" id="${AdminNutrientsSelect.nutrientsID}" title="${AdminNutrientsSelect.nutrientsName} 수정"

onclick="location.href='admin?cmd=AdminNutrientsDetail&nutrientsID=${AdminNutrientsSelect.nutrientsID}'" >

<td>${ AdminNutrientsSelect.nutrientsID }</td>

<td>${ AdminNutrientsSelect.nutrientsName }</td>

<td>${ AdminNutrientsSelect.nutrientsPrice }</td>

<td>${ AdminNutrientsSelect.nutrientsCategory }</td>

<td><img src="${ AdminNutrientsSelect.nutrientsIMG }" style="width: 100px; height: 100px"/></td>

<td>${ AdminNutrientsSelect.nutrientsDetail }</td>

<td>${ AdminNutrientsSelect.nutrientsDOM }</td>

<td>${ AdminNutrientsSelect.dailyInTake }</td>

<td>${ AdminNutrientsSelect.numperbottle }</td>

<td>${ AdminNutrientsSelect.remainingNum }</td>

<td>${ AdminNutrientsSelect.link }</td>

</tr>

</c:forEach>

이후 목록에서 수정을 누르면 'admin?cmd=AdminNutrientsDetail&nutrientsID=${AdminNutrientsSelect.nutrientsID}'로 컨트롤러를 호출 이후 수정 사항을 진행하고

<update id=*"AdminNutrientsUpdate"*

parameterType=*"com.hwf.model.NutrientsDTO"*>

update nutrients set

nutrientsName = #{nutrientsName},

nutrientsPrice = #{nutrientsPrice},

nutrientsCategory = #{nutrientsCategory},

nutrientsImg = #{nutrientsIMG},

nutrientsDetail = #{nutrientsDetail},

nutrientsDom = to\_date(#{nutrientsDOM},'yyyy-mm-dd'),

dailyIntake = #{dailyInTake},

numPerBottle = #{numperbottle},

remainingNum = #{remainingNum},

link = #{link}

where nutrientsId = #{nutrientsID}

</update>

Update 문을 dao로 호출하여 사용할 수 있다.

이후 기능등은 시간 상 생략한다.

제5절 결론 및 개선방향

5. 1 결론 및 기대효과

이번 프로젝트는 Webservlet 형태의 게시판 예제를 토대로 논의를 하고 복습을 중점적으로 하는 것에 목적을 두었다. 프로젝트 작업에 있어 팀원들마다 기능과 역할을 분담하였으며, 매일 오전에 회의를 통하여 팀원들 간의 작업 진행속도를 공유하였습니다. 각자의 기능 구현에 있어 문제가 되는 부분들은 소통을 통해 구현 방법 수정을 하거나 해당 기능 구현을 아는 팀원이 도와주며, 목표한 기능 구현까지 원활하게 진행하고자 하였으며, 수시로 branch에 작업 한 것을 Git에 올렸습니다. 그리고 회의록 및 추가 작업 관련 내용을 정리하여 Notion에 올려 본인 뿐만 아니라 팀원들 또한 누가 어떤 작업을 진행했는지 기재하였습니다.

팀 프로젝트를 통하여 정규표현식를 활용한 아이디 조건 생성, 중복 아이디 체크와 본인인증 서비스를 활용한 로그인 기능은 다른 프로젝트에서 로그인 기능을 다채롭게 사용할 수 있게 되었다. Admin 기능에 회원조회, 상품관리 및 등록, 주문조회, 고객 QnA를 통해 다른 프로젝트에 관리 프로그램에 있어 이를 적용할 수 있게 되었다. My Page기능에서 개인정보뿐만 아니라 설문조사 및 결과 확인, 인바디 등록 및 결과 확인, 구매 내역 확인, QnA(1:1) 문의는 고객 스스로 정보를 확인을 하고 문의를 통해 답변을 받을 수 있게 하였으며, 설문, 인바디 결과에 따른 추천 상품 매칭과 노출된 상품 또는 검색한 상품의 내용을 상세히 확인. 장바구니에 원하는 상품을 담으며, 영양제의 예약구매기능을 통해 구매 날짜를 별도 설정과 카드사에 따른 결제 방법 등을 통하여 다른 온라인 이커머스 사이트에서도 이를 적용할 수 있게 되었습니다.

5. 2 문제점 및 개선 사항

<문제점>

1. 전체 및 각 상품 페이지 및 1:1 문의 게시글 페이징 미구현

- 상품 리스트를 확인하거나 게시글 등을 확인 할 때 페이징 처리가 되지않아 전체 검색의 경우 모든 항목이 출력되는 상황이 발생되며, 대량의 데이터를 불렀을 경우 한 페이지에 전체적으로 노출되어 사용자가 사용하는데 있어 불편함을 초래할 수 있다.

→ 리스트 출력과 다른 기능들과의 적용을 우선 순위로 하였으며, 페이징 처리를 차순위로 설정하였다. 시간을 추가로 부여하게 된다면 페이징 처리를 적용하여 전반전인 조회에서의 구분하는 것을 개선하고자 한다.

2. 1:1 문의 기능에 파일 업로드 미구현

- 1:1 문의에 파일 업로드 첨부가 되지않아 기재되어 있는 글로는 고객 클레임에 대하여 정확한 이해를 하기 어려워 의도를 파악하는데 있어 한계가 있다.

→ 고객 문의 글 작성과 관리자가 답변할 수 있는 형태로 만드는 것을 우선적으로 하였으며, 마감기한 내 공부해서 진행하는데 늦을 수 있어 구현하지 못 했습니다. 추가로 공부하게 된다면 스프링 웹 프로젝트 p149, p.487 파일 업로드 관련 내용을 참고하여 구현하도록 하겠습니다.

<개선사항>

1. Ajax 사용

- Webservle를 사용하는 팀 프로젝트에서 Ajax를 적용을 하는 방법을 찾고 적용하는데 시간소요가 있어 Ajax를 적용을 하지 못 했다. 시간이 추가적으로 부여된다면 Ajax를 적용하여 웹페이지 작업을 조금 더 수월하게 작업하게 만들고 싶다.

2. 관리자 주문조회

- 관리자 주문조회시 전체조회만 되어 식별하는데 아쉬운 점들이 있다. 따라서 시간이 더 주어진다면 상품 품목별로 주문 리스트를 출력하게 만들고 싶다.

3. 관리자 재고현황

- 재고 현황을 만들 수 있었으나 해당 기능은 차순위로 미뤘으며, 구현 전 최종 Merge 작업 및 부족한 CSS 작업을 우선으로 진행하게 되어 구현하지 못하였다. 시간이 더 있었다면 구현을 할 수 있다.

4. 리뷰

- 상품 별 리뷰 내용 조회 및 상품을 구매한 사용자만 그 상품에 대한 리뷰를 작성할 수 있게 하였으나 구현이 늦어 해당 프로젝트에서는 적용하지 못 했다. 적용하는 것에 있어 가장 어려운 이유로는 상품 별 리뷰 내용이 출력되는 과정에서 리뷰의 유무에 따라 리뷰가 나오게 하였으나 주문 기능과 충돌하는 문제 때문이며, 시간이 더 주어진다면 해당 문제 해결 및 파일 업로드, 페이징 처리를 포함하여 구현하려고 한다.