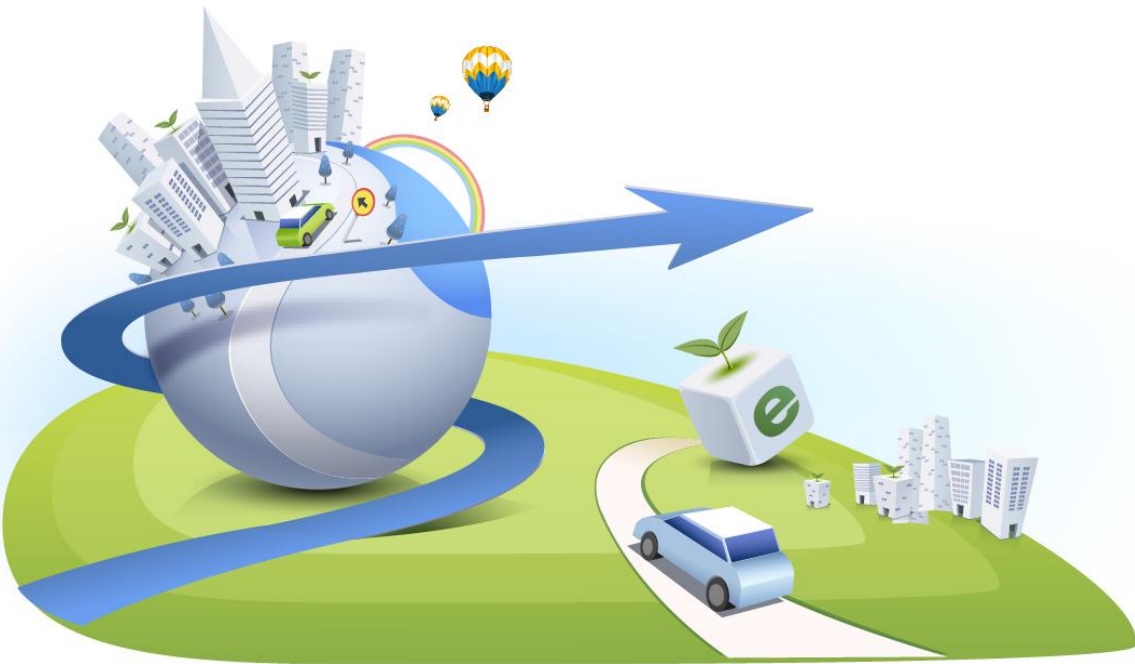


2017. 11.13~ 2017. 12.15 | KOSTA 163th

## Java 개발자 양성 과정

### 함께 하시 개 -육남매-

반려견과 견주들의 오프라인 모임을 위한  
모임 개설, 참여와 커뮤니티 서비스를 제공하는 통합 솔루션




## 목차

1	프로젝트 소개 .....	3
1.1	프로젝트 요약서 .....	3
1.2	서비스화면 .....	4
2	개발환경 및 적용기술 .....	5
2.1	개발환경 .....	5
2.2	적용기술 .....	6
3	프로젝트 관리 .....	7
3.1	마일스톤 .....	7
3.2	칸반 .....	8
4	분석설계 .....	9
4.1	Use Case Diagram .....	9
4.2	Use Case 시나리오 .....	10
4.3	기능명세서 .....	12
4.4	ERD .....	16
4.5	Class Diagram .....	17
5	주요기능 .....	19
6	평가 .....	29
6.1	Best Practice .....	29
6.2	Lesson Learned .....	30

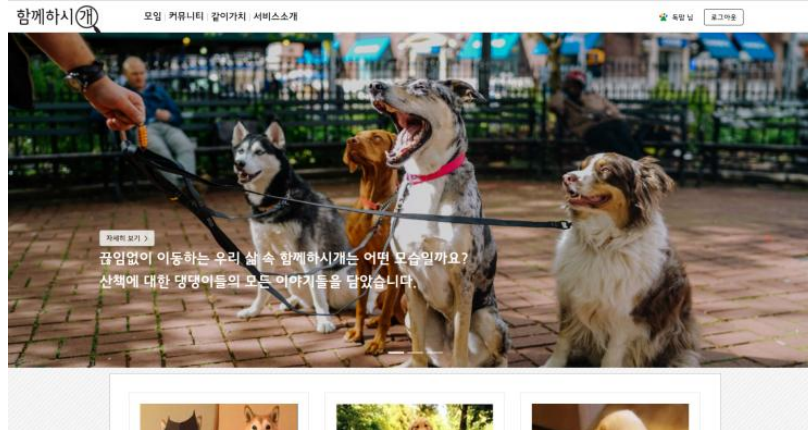
## 1 프로젝트 소개

### 1.1 프로젝트 요약서

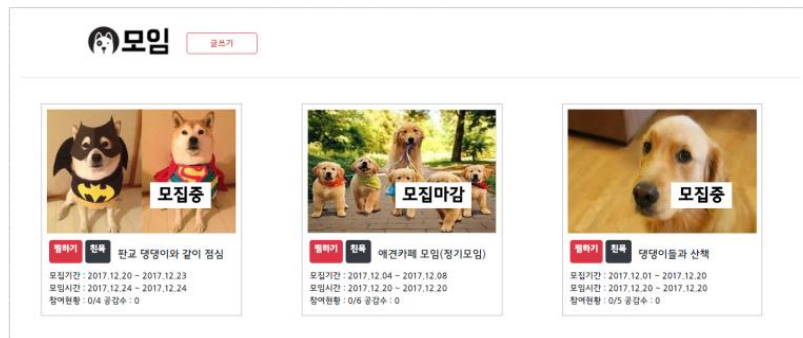
프로젝트	KOSTA 163기 Final Project	
개발기간	2017.11.13 ~ 2017.12.15	
서비스	이름	함께하시개 
	개요	반려견과 견주들의 오프라인 모임을 위한 모임 개설, 참여와 커뮤니티 서비스를 제공하는 통합 솔루션
서비스 내용	<p>함께 하시 개는 반려견 1000만 시대에 발맞춰 반려견과 견주들이 더불어 살 수 있는 반려견 문화를 만들기 위한 서비스이다.</p> <p>반려견의 산책이나 친목, 견주의 세미나 등 다양한 성격의 모임을 개설하고 참여, 공감하는 모임에 관련된 서비스가 주된 내용이다.</p> <p>그 외에 '커뮤니티'의 여러 게시판을 통하여 다양한 정보를 교류할 수 있으며 '같이가치'라는 기부 서비스를 제공하여 어려움을 겪고 있는 반려견과 견주에게 도움을 줄 수 있는 서비스이다.</p>	
서비스 목적	<ul style="list-style-type: none"> <li>■ 반려견의 사회성 신장</li> <li>■ 반려견주의 교류 모임 조성</li> <li>■ 반려견주들이 서로 다양한 정보를 주고 받을 수 있는 커뮤니티 조성</li> <li>■ 반려견들을 위한 기부 서비스 제공</li> </ul>	
개발 내용	Spring Model 2 MVC 패턴을 적용한 웹서비스 개발	
프로젝트 목적	<ul style="list-style-type: none"> <li>■ 현재까지 배운 지식을 프로젝트에 적용하여 총 복습하는 기회를 갖는다.</li> <li>■ Spring Framework를 적용하여 프로그래밍에 대한 실전감각을 키운다.</li> <li>■ 팀원들간의 소통을 바탕으로 협의 능력을 키운다.</li> </ul>	
적용기술	S/W Architecture	Spring MVC
	Language	Java
	Web	Servlet/JSP, HTML, CSS,
	Framework	Spring Framework, Tiles, MyBatis, Bootstrap

## 1.2 서비스화면

### 메인화면



### 모임 목록 화면



### 모금함 화면










## 2 개발환경 및 적용기술

### 2.1 개발환경

o/s		윈도7, 2009년 10월 22일 출시, 마이크로소프트( <b>MS</b> )가 개발한 운영체제( <b>OS</b> )
JavaEE		자바 개발 키트(Java Development Kit, JDK)는 자바 SE, 자바 EE, 또는 자바 ME 플랫폼 중 하나를 구현한 것으로 솔라리스, 리눅스, 맥 OS X, 또는 윈도 자바 개발자를 대상으로 오라클에 의해 바이너리 제품으로 제공된다.
UML		요구분석, 시스템설계, 시스템 구현 등의 시스템 개발 과정에서, 개발자간의 의사소통을 원활하게 이루어지게 하기 위하여 표준화한 모델링 언어
Database		영국의 Oracle사에서 개발하여 판매하는 관계 데이터 베이스 관리 시스템
WAS		아파치 톰캣 (Apache Tomcat)은 아파치 소프트웨어 재단에서 개발된 서블릿 컨테이너(또는 웹 컨테이너)만 있는 웹 애플리케이션 서버
Page Framework		웹 페이지의 상단이나 하단, 메뉴와 같은 반복적으로 사용되는 부분들에 대한 정보를 한 곳에 모아 둔 프레임워크.
형상관리		Git은 프로그램 등의 소스 코드 관리를 위한 분산 버전 관리 시스템이다. 빠른 수행 속도에 중점을 두고 있는 것이 특징이다.
IDE		프로그래밍을 하려면 코드를 작성하고, 저장하고 컴파일 및 디버깅을 도와주는 통합 개발 환경(Integrated Development Environment, IDE)이 필요하다.

## 2.2 적용기술

Language		<p>객체 지향 프로그래밍 언어로서 보안성이 뛰어나며 컴파일한 코드는 다른 운영 체제에서 사용할 수 있도록 클래스(class)로 제공된다.</p>
WEB		<p>서버에서 웹페이지 등을 동적으로 생성하거나 데이터 처리를 수행하기 위해 자바(Java)로 작성된 프로그램을 말한다.</p>
		<p>HTML 웹 문서를 만들기 위하여 사용하는 기본적인 프로그래밍 언어의 한 종류이다. 하이퍼텍스트를 작성하기 위해 개발됨.</p> <p>CSS 웹 문서의 전반적인 스타일을 미리 저장해 둔 스타일시트. 문서 전체의 일관성을 유지할 수 있고, 세세한 스타일 지정의 필요를 줄어줄게 함.</p>
		<p>자바스크립트는 크로스 플랫폼(cross platform), 객체지향 스크립트 언어로 웹페이지의 동작을 담당</p>
		<p>비동기식 자바스크립트 XML(Asynchronous Javascript And XML)의 약자. 하이퍼텍스트 표기언어(HTML)만으로 어려운 다양한 작업을 웹페이지에서 구현해 이용자가 웹페이지와 자유롭게 상호 작용할 수 있도록 하는 기술</p>
		<p>jQuery (제이쿼리)는 HTML의 클라이언트 사이드 조작을 단순화 하도록 설계된 크로스 플랫폼의 자바스크립트 라이브러리다.</p>
		<p>자바서버 페이지 표준 태그 라이브러리 (JavaServer Pages Standard Tag Library, 약칭 JSTL )은 Java EE 기반의 웹 애플리케이션 개발 플랫폼을 위한 컴포넌트 모음이다.</p>

### 3 프로젝트 관리

#### 3.1 마일스톤<sup>1</sup>

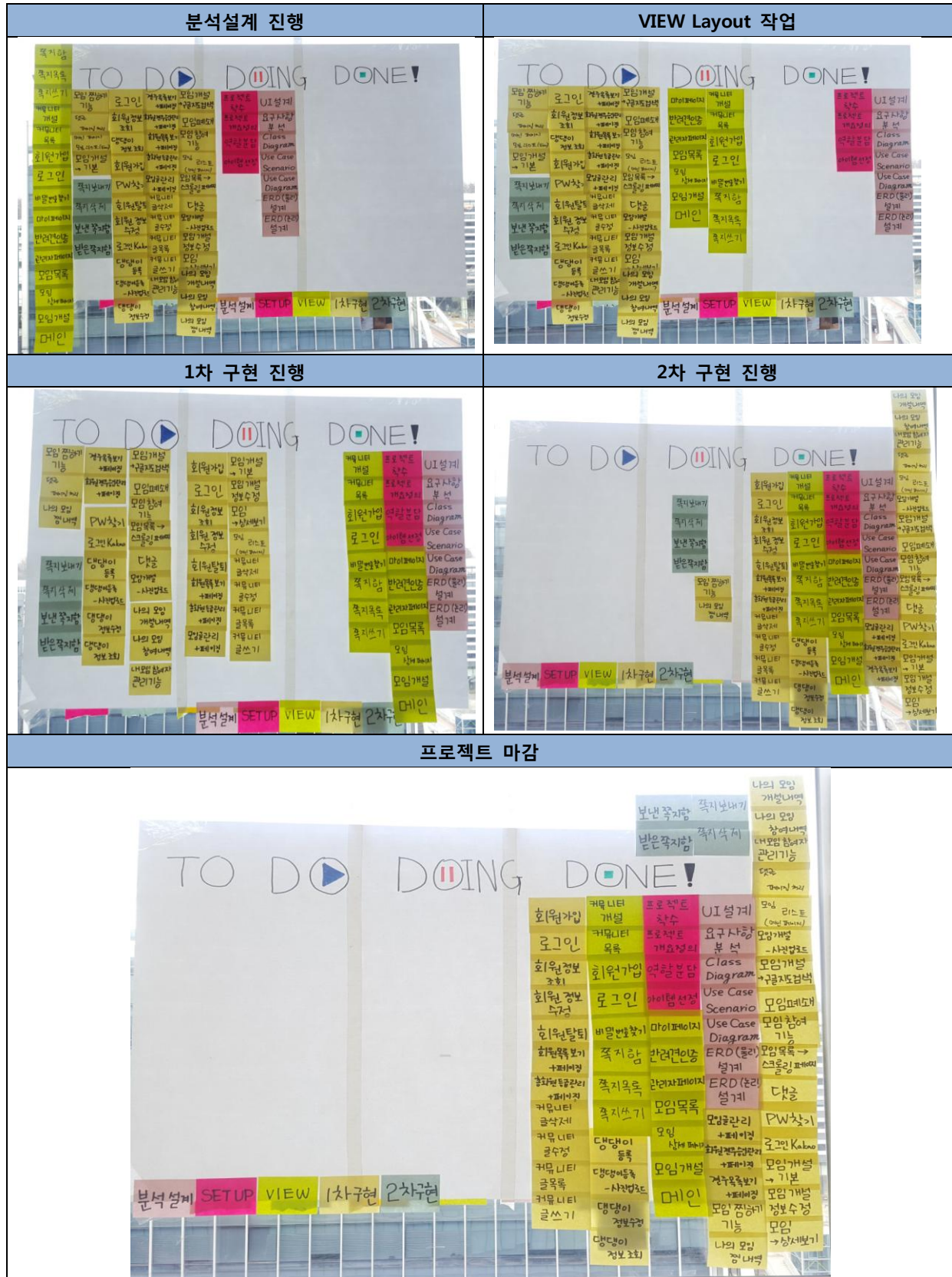
마일스톤																
	세부항목	schedule														
		11/13	11/14	11/15	11/16	11/17	11/18	11/20	11/21	11/22	11/23	11/24	11/25	11/27	11/28	11/29
프로젝트 기획	프로젝트 착수															
	아이템선정															
분석/설계	요구사항 분석															
	UseCase Diagram															
	UseCase Scenario															
	UI 설계															
	ERD 설계															
	Class Diagram 설계															
	개발환경 구축															
구현 1차	로그인 및 로그아웃															
	회원가입															
	모임 리스트															
	모임 개설															
	모임 부가기능															
	마이페이지															
	관리자 페이지															
	커뮤니티 리스트															
	커뮤니티 카테고리 관리															

	세부항목	schedule													
		11/30	12/01	12/02	12/04	12/05	12/06	12/07	12/08	12/09	12/11	12/12	12/13	12/14	12/15
중간 테스트	회원 기능 테스트														
	관리자 기능 테스트														
	모임 기능 테스트														
	커뮤니티 기능 테스트														
구현 2차	쪽지함 개설														
	쪽지 리스트														
	모임 찜 기능														
	같이가기 리스트														
	같이가기 개설														
최종 테스트	쪽지 기능 테스트														
	같이가기 기능 테스트														
	통합 테스트														
산출물 정리	결과 보고서 작성														
	동영상 촬영														
	프로젝트 마감														

<sup>1</sup> 마일스톤 ( Milestone ) : 프로젝트 진행 과정에서 특기할 만한 사건이나 이정표를 말한다. 프로젝트 일정관리를 위해 반드시 필요한 지점을 체크하기 위해 사용한다



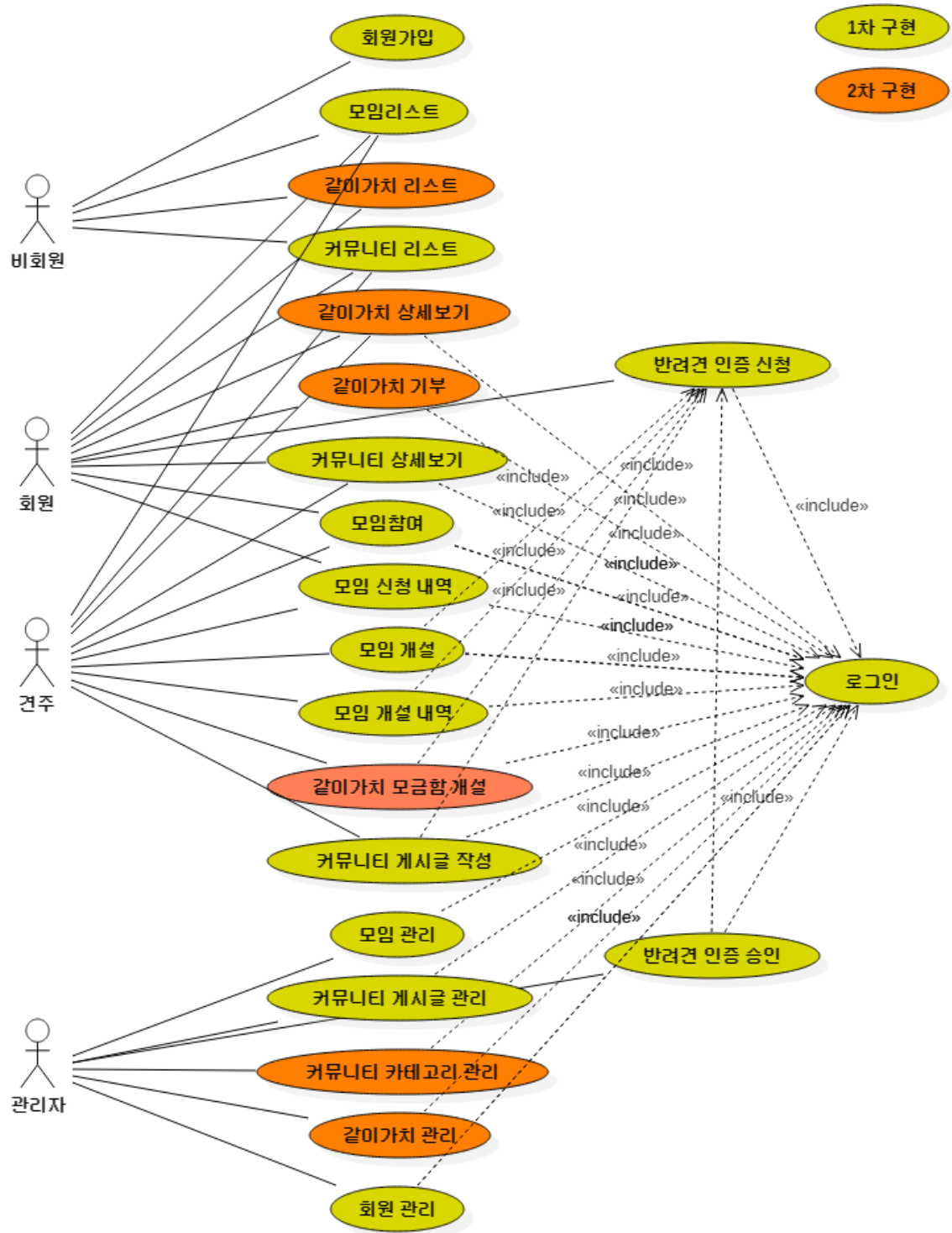
## 3.2 칸반





## 4 분석설계

### 4.1 Use Case Diagram



## 4.2 Use Case 시나리오

<b>유스케이스명</b>	<b>반려견 인증 신청</b>
<b>관련 액터</b>	회원
<b>시나리오 개요</b>	회원의 권한을 가진 사용자가 견주의 권한을 얻기 위해 신청할 때 호출 된다.
<b>사전조건</b>	'함께 하시 개' 서비스에 가입 하여야 한다.
<b>메인 시나리오</b>	<ol style="list-style-type: none"> <li>1. '마이페이지' 버튼을 클릭 한다.</li> <li>2. 반려견의 사진을 업로드 한다.</li> <li>3. 반려견의 상세정보를 입력 한다.</li> <li>4. '반려견 인증 신청' 버튼을 클릭 한다.</li> </ol>

<b>유스케이스명</b>	<b>반려견 인증 승인</b>
<b>관련 액터</b>	관리자
<b>시나리오 개요</b>	회원의 권한을 가진 사용자의 권한을 견주로 변경할 때 호출 된다.
<b>사전조건</b>	서비스에 가입한 회원이 '반려견 인증 신청'을 해야 한다.
<b>메인 시나리오</b>	<ol style="list-style-type: none"> <li>1. 관리자가 '관리하기&gt;회원관리&gt;견주 대기자' 버튼을 클릭한다.</li> <li>2. 견주 대기자들이 작성한 댕댕이의 정보를 검토한다.</li> <li>3. 검토한 내용이 적합할 시 회원(견주 대기자)의 권한을 견주로 변경 한다.</li> </ol>

<b>유스케이스명</b>	<b>회원 관리</b>
<b>관련 액터</b>	관리자
<b>시나리오 개요</b>	'함께 하시 개' 회원들을 관리할 때 호출 된다.
<b>사전조건</b>	'함께 하시 개' 서비스에 가입한 회원이 존재 한다.
<b>메인 시나리오</b>	<ol style="list-style-type: none"> <li>1. 관리자가 '관리하기&gt;회원관리' 버튼을 클릭한다.</li> <li>2. 전체회원, 견주, 견주 신청자 회원들의 목록을 확인 한다. <ol style="list-style-type: none"> <li>2-1. 회원의 등급 <ul style="list-style-type: none"> <li>- 회원</li> <li>- 견주</li> <li>- 탈퇴회원</li> <li>- 카페 운영자</li> <li>- 시스템 관리자</li> </ul> </li> </ol> </li> <li>3. 관리자는 전체회원 목록에서 회원들의 등급을 변경 가능하다</li> </ol>

<b>유스케이스명</b>	<b>모임 참여</b>
<b>관련 액터</b>	회원, 견주
<b>시나리오 개요</b>	개설된 모임에 참여를 원할 때 호출되는 유스케이스
<b>사전조건</b>	개설된 모임에 한해서 참여를 할 수 있다.
<b>메인 시나리오</b>	<ol style="list-style-type: none"> <li>1. 모임리스트에서 참여를 원하는 모임에 참여 신청 버튼을 클릭한다.</li> <li>2. 참여를 원하는 사람은 자신의 정보를 입력한다.</li> <li>3. 모임 참여를 완료하면, 마이페이지의 신청한 모임 내역에서 자신이 신청한 모임의 목록을 볼 수 있다.</li> </ol>

<b>유스케이스명</b>	<b>모임 개설</b>
<b>관련 액터</b>	견주
<b>시나리오 개요</b>	모임을 개설할 때 호출된다.
<b>사전조건</b>	견주 인증을 받은 회원이 이용 가능하다.
<b>메인 시나리오</b>	<ol style="list-style-type: none"> <li>1. 견주는 모임 개설 버튼을 클릭한다.</li> <li>2. 견주에게 모임 개설 폼이 제공된다.</li> <li>3. 견주는 모임명, 모임 일시, 모임 장소, 모임 소개 등 모임에 대한 정보를 입력한다.</li> <li>4. 모임 개설을 완료하면, 모임리스트에 개설된 모임이 올라간다.</li> <li>5. 마이페이지의 개설한 모임 내역에서 자신이 개설한 모임의 목록을 볼 수 있다.</li> </ol>

### 4.3 기능명세서

주요기능	세부기능	기능설명	구현상세
로그인	회원가입	<ul style="list-style-type: none"> <li>- 회원이 아닌 사용자에게 제공</li> <li>- 회원가입 시 유효성검사(중복확인 등) 적용</li> </ul>	<ul style="list-style-type: none"> <li>- MemberService isIdCheck() isNickCheck() registerMember()</li> <li>- MemberDAO isIdCheck() isNickCheck() registerMember() registerRole()</li> </ul>
	비밀번호 찾기	<ul style="list-style-type: none"> <li>- 비밀번호를 잊은 회원에게 제공</li> <li>- 비밀번호 재설정 가능</li> </ul>	<ul style="list-style-type: none"> <li>- MemberService isIdPwAnswer()</li> <li>- MemberDAO isIdPwAnswer()</li> </ul>
	마이페이지	<ul style="list-style-type: none"> <li>- 회원에게 자신의 정보를 열람 및 수정 시 사용</li> <li>- 개인정보 수정 및 회원탈퇴 가능</li> </ul>	<ul style="list-style-type: none"> <li>- MemberService findMyPageInfoById()</li> <li>- MemberDAO findMyPageInfoById()</li> </ul>
	카카오 로그인 연동	<ul style="list-style-type: none"> <li>- 카카오 아이디를 통해 로그인 가능</li> </ul>	<ul style="list-style-type: none"> <li>- 카카오 API 이용</li> </ul>
	반려견 인증	<ul style="list-style-type: none"> <li>- 일반 회원이 사용</li> <li>- 댕댕이맘으로 승급하기 위해 반려견 인증</li> </ul>	<ul style="list-style-type: none"> <li>- MemberService registerDogInfo()</li> <li>- MemberDAO registerDogInfo()</li> </ul>
모임	모임 개설	<ul style="list-style-type: none"> <li>- 댕댕이맘으로 인증된 회원이 모임 개설</li> </ul>	<ul style="list-style-type: none"> <li>- MeetingService registerMeeting()</li> <li>- MeetingDAO registerBoard() registerGatheringdate() registerImg() registerMeeting() registerMeetingType()</li> </ul>

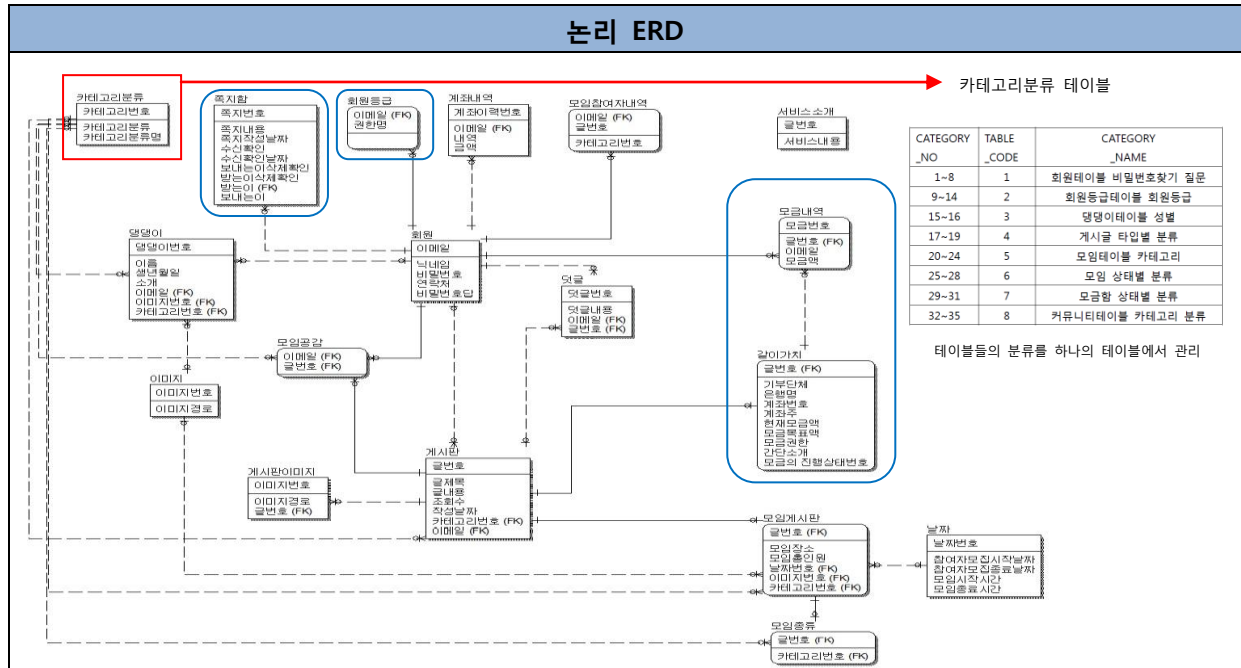
	모임 참여	<ul style="list-style-type: none"> <li>- 댕댕이맘으로 인증된 회원이 모임 참여 및 참여 취소</li> <li>- 개설자는 참여 불가</li> </ul>	<ul style="list-style-type: none"> <li>- MeetingService addAttenderMember() removeAttenderMember() removeReply()</li> <li>- MeetingDAO addAttenderMember() isAttender() removeAttenderMember() removeReply()</li> </ul>
	모임 관리	<ul style="list-style-type: none"> <li>- 모임을 개설한 개설자가 모임 관리 : 모임 폐쇄, 모임 수정, 모임 참여자 관리</li> <li>- 모임 참여자가 있을 시 모임 수정 불가 (모임 폐쇄는 가능)</li> </ul>	<ul style="list-style-type: none"> <li>- MeetingService getMyMeetingAttender() removeMeetingInfo() removeReply() removeAttenderByFounder()</li> <li>- MeetingDAO removeBoardByBoardNo() removeGatheringDateByBoardNo() removeImgByBoardNo() removeReply() removeAttenderMember() removeAttenderByFounder()</li> </ul>
쪽지	쪽지 보내기	<ul style="list-style-type: none"> <li>- 회원간 쪽지 보내기 및 쪽지 받기 가능</li> <li>- 읽지 않은 쪽지 표시</li> </ul>	<ul style="list-style-type: none"> <li>- MessageService registerMessage() findSendMessageByNo() findReceiveMessageByNo() removeSendMessage() removeRecieveMessage()</li> <li>- MessageDAO isMessageReadChk() registerMessage() findSendMessageByNo() findReceiveMessageByNo() setSenderDFlag() setRecieverDFlag() removeSendMessage() removeRecieveMessage()</li> </ul>

커뮤니티	게시글 작성	- 회원이 커뮤니티 게시판을 통해 게시글 작성	<ul style="list-style-type: none"> <li>- CommunityService registerCommunityInfo() removeCommunityInfo() updateCommunityInfo()</li> <li>- CommunityDAO registerCommunityInfo() removeCommunityInfo() updateCommunityInfo()</li> </ul>
같이가기	기부	<ul style="list-style-type: none"> <li>- 회원이 개설된 모금함에 기부</li> <li>- 기부 시에 결제 모듈 사용</li> <li>- 카카오페이 API를 통해 결제</li> </ul>	<ul style="list-style-type: none"> <li>- DonationService addDonation()</li> <li>- DonationDAO deposit() withdraw() payHistory() patDonation()</li> </ul>
	모금함 개설	<ul style="list-style-type: none"> <li>- 회원이 모금함을 개설</li> <li>- 관리자의 승인을 받아 모금함 글 게시</li> </ul>	<ul style="list-style-type: none"> <li>- DonationService registerDonation()</li> <li>- DonationDAO registerImg() registerMotherDonation() registerDonation()</li> </ul>
관리자	회원 관리	<ul style="list-style-type: none"> <li>- 회원의 등급 조절 가능</li> <li>- 댕댕이맘 대기자 승인 및 거절 가능</li> <li>- 탈퇴회원 관리</li> </ul>	<ul style="list-style-type: none"> <li>- AdminService getAllMemberList() getAllTierList() removeManagerDogmom() setMemberTier() setManagerDogmomPermit()</li> <li>- AdminDAO getAllMemberList() getAllTierList() removeMemberTier() registerTierMember() setMemberTier() setManagerDogmomPermit()</li> </ul>



	커뮤니티 게시물 관리	- 커뮤니티 게시물 폐쇄 가능	- AdminService getAllCommunityList()  - AdminDAO getAllCommunityList()
	커뮤니티 카테고리 관리	- 커뮤니티 카테고리 추가 및 삭제	- AdminService registerAddCommunityType() removeCommunityType()  - AdminDAO registerAddCommunityType() removeCommunityType() removeCommunityTypeBoard()
	같이가치 관리	- 개설 신청한 모금함을 승인 및 거절	- AdminService getAllDonationList() getDonationListPerState() setDonationStateChange()  - AdminDAO getAllDonationList() getDonationListPerState() setDonatiuonStateChange()
	모임 관리	- 개설된 모임을 폐쇄	- AdminService getAllMeetingList()  - AdminDAO getAllMeetingList()

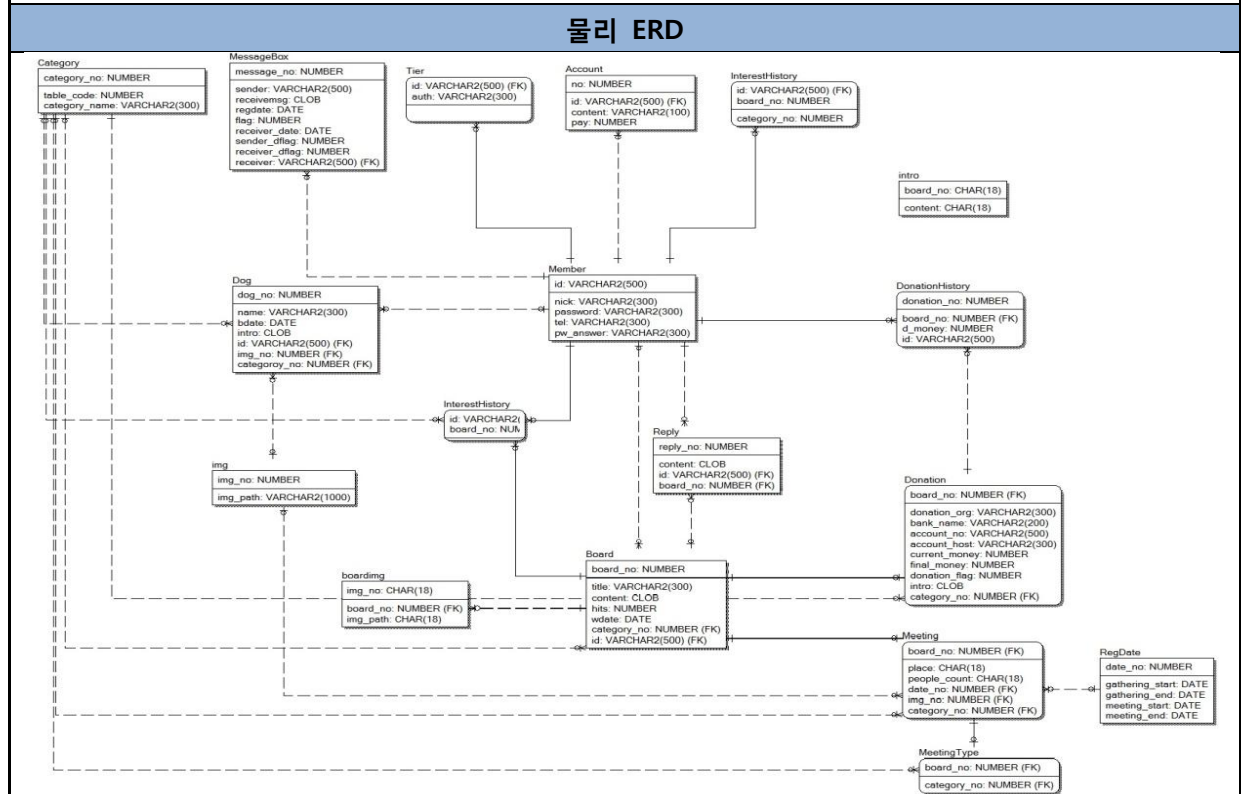
## 4.4 ERD



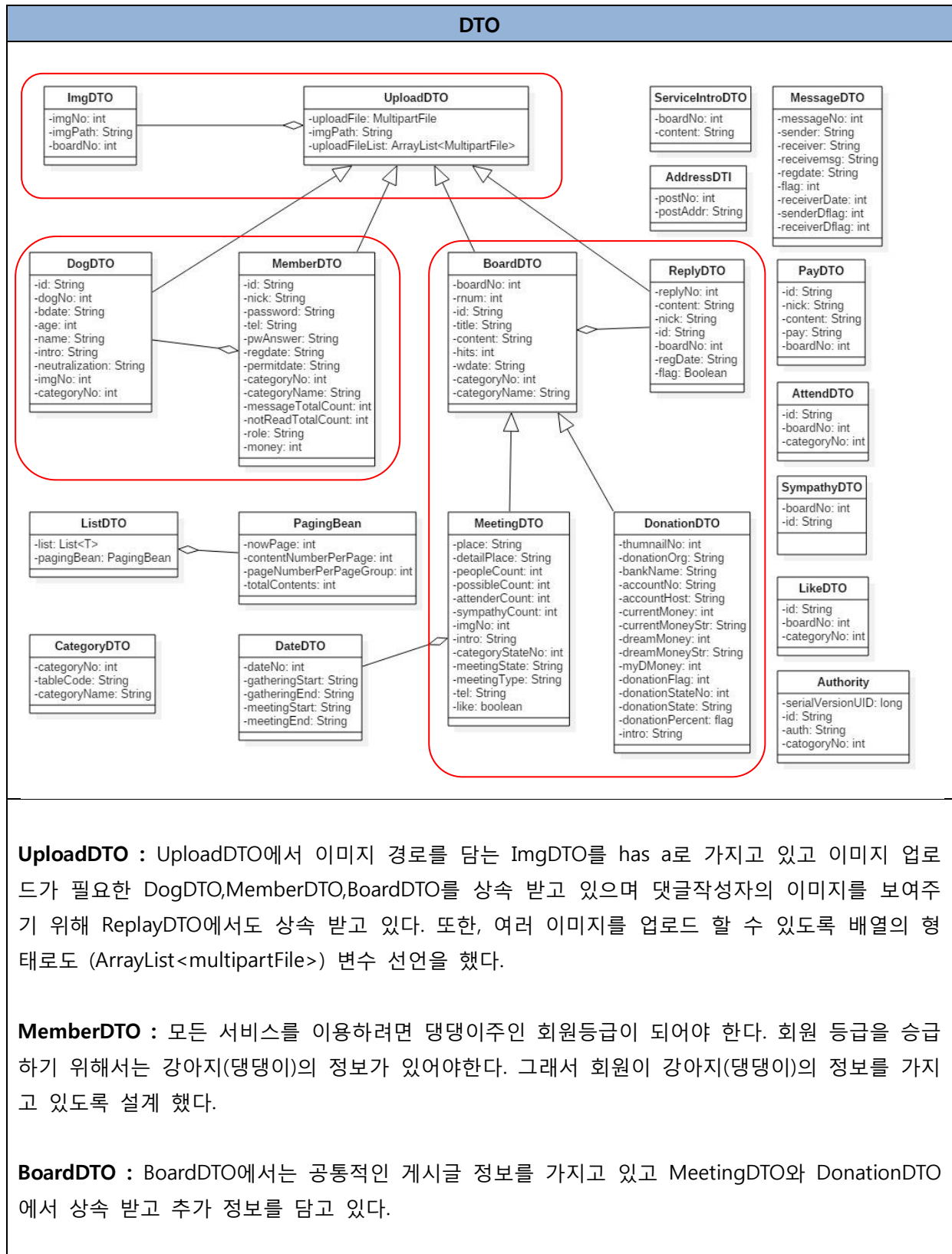
**쪽지함 테이블** : 다수회원이 하나의 쪽지함을 가지고 있다. 또한 보낸 이가 쪽지를 보낸 후 삭제 하더라도 받는 이가 확인할 수 있도록 '삭제확인' 컬럼을 추가 했다.

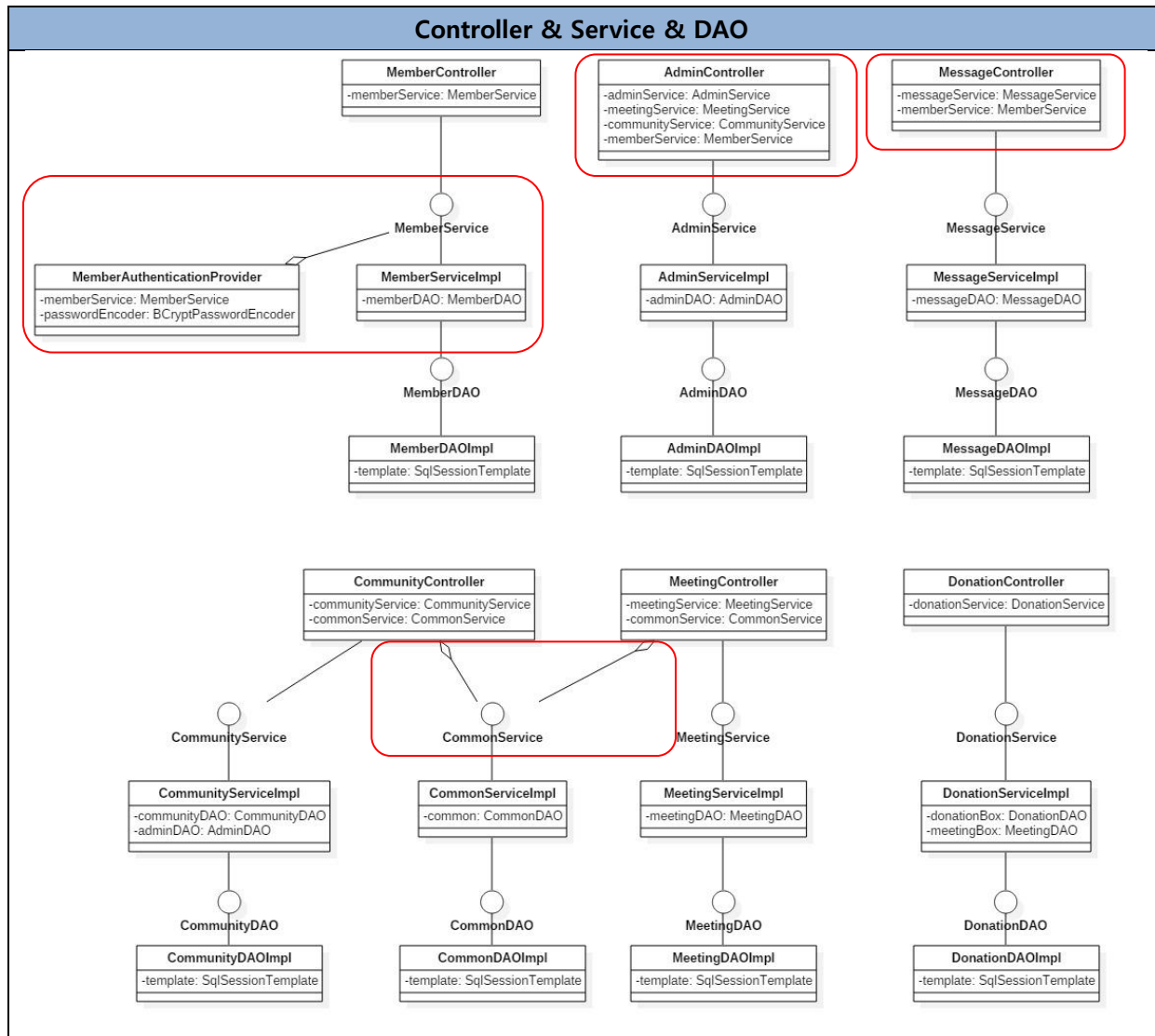
**회원등급 테이블** : 회원 등급테이블은 회원테이블 PK이와 권한명을 복합 PK로 설정했다.

**같이가치 테이블 & 모금내역 테이블** : 모금함 서비스를 하기 위해 하나의 게시글 당 모금내역을 가지고 있다.



## 4.5 Class Diagram





**AdminController** : AdminController에서는 회원과 게시글들을 관리하기 위해 AdminService, MemberService, MeetingService, CommunityService를 has a 관계로 표현했다.

**MemberAuthenticationProvider** : Spring Security 적용하기 위해 MemberService를 has a로 가지고 있다.

**MessageController** : 메시지 기능 수행 시, 보낼 회원 정보를 가져오기 위해 MemberService를 has a로 표현했다. MemberService의 findMemberById 메소드를 사용하여 회원의 정보를 가져온다.

**CommonService** : 공통기능을 수행하는 Service이다.

조희수는 MemtingController와 CommunityController의 공통기능으로 has a로 표현했다.

## 5 주요기능

SQL	OUTER JOIN
	LISTAGG
CODE	Image Upload
	Spring Security
기능	커뮤니티 카테고리
	모임 참여자 관리
OPEN API	Smart Editor
	Daum Maps
	Kakao Pay
	Kakao Login

## SQL

### 1)OUTER JOIN

#### Service Content :

회원이 작성한 댓글 내용을 DB에서 가져올 때, Reply 테이블과 Member테이블의 정보를 함께 가져와야 합니다. 동시에 회원의 프로필 사진도 같이 보여줘야 하는데 프로필 사진이 존재하지 않을 경우에도 댓글의 내용을 가져와야 합니다. 그러나 이 경우 일반적인 Join 으로는 불가능하고 이를 해결하기위해 Outer join을 이용 하였습니다.

#### SQL :

```
SELECT
  r.reply_no as replyNo, r.content, to_char(r.reg_date, 'yyyy-mm-dd hh24:mi')
  as regDate, r.id, m.nick, i.img_path as imgPath
FROM
  (select img_path, id from img i, dog d where i.img_no=d.img_no) i, reply r,
  member m
WHERE
  r.board_no=#{value} and m.id=r.id and i.id(+)=r.id order by r.reply_no desc
```

### 2)LISTAGG

#### Service Content :

‘같이 가치’ 서비스에서 ‘모금함 개설’ 시 회원은 여러 개의 다수 이미지를 등록하여야 합니다. 이 때, 이미지의 정보는 Img 테이블에 저장되게 되고 이미지는 회원 아이디, 글 번호, 이미지 번호 등을 하나의 Row로 저장하게 됩니다. 이미지 테이블에서는 글 번호에 해당하는 여러 개의 정보를 하나의 게시글에서 보여줘야 하고, 이런 제약조건의 글을 조회하기 위해서 LISTAGG와 GROUP BY를 같이 사용하였습니다.

#### SQL :

```
SELECT
  dona.board_no as boardNo, dona.donation_org as donationOrg, dona.bank_name
  as bankName, dona.account_no as accountNo, dona.account_host as accountHost,
  dona.current_money as currentMoney, dona.final_money as dreamMoney,
  dona.donation_flag as donationFlag, dona.category_no as categoryNo,
  dona.img as imgPath, bo.title, bo.content, bo.hits, bo.wdate, bo.id
FROM
  (select d.board_no, donation_org, bank_name, account_no, account_host,
  current_money, final_money, donation_flag, category_no, b.img
  FROM donation d, (SELECT board_no,LISTAGG(img_path,',') WITHIN GROUP (ORDER
  BY board_no) as img FROM boardimg GROUP BY board_no) b
WHERE
  d.board_no = b.board_no and d.board_no = #{value}) dona, board bo WHERE
  dona.board_no = bo.board_no
```



## CODE

### 1)Image Upload

#### Service Content :

1. 땡땡이 정보등록, 모임개설, 모금개설 등의 서비스에서 회원은 이미지 업로드를 할 수 있습니다.
2. 글의 등록정보는 <T> 라는 타입의 값을 참조하게 되는데 이 때, <T>는 Generic으로 해결 하였습니다.

#### CODE :

```
public void setImageUpload(HttpServletRequest request, T dto) {
    uploadPath =
        request.getSession().getServletContext().getRealPath("/imgPath/");
    File uploadDir = new File(uploadPath);
    if (uploadDir.exists() == false)
        uploadDir.mkdirs();

    MultipartFile file = dto.getUploadFile();

    if (file != null && file.isEmpty() == false) {
        dto.setImgPath(file.getOriginalFilename());
        File uploadFile = new File(uploadPath + file.getOriginalFilename());
        try {
            file.transferTo(uploadFile);
        } catch (IllegalStateException | IOException e) {
            e.printStackTrace();
        }
    }
}
```

### 2)Spring Security

#### Service Content :

1. '함께 하시 개' 서비스에 가입한 회원들은 일종의 권한(Auth)를 얻게 됩니다.
2. 권한에 따라 접근 가능범위를 설정함으로써 보안성을 높입니다.
3. 메서드 상단부에 Security 를 명시, AOP적인 선언방식으로 접근가능범위를 설정합니다.

#### XML : Spring Security 인증 처리 로직을 정의한 클래스

```
<bean name="customizedAuthenticationProvider"
      class="com.sixfam.withpet.security.MemberAuthenticationProvider" />

<sec:authentication-manager>
    <sec:authentication-provider ref="customizedAuthenticationProvider"/>
</sec:authentication-manager>
```

#### CODE : Controller 상단부에 선언

```
@Secured("ROLE_MEMBER")
```

## 기능

### 1)커뮤니티 카테고리

#### Service Content :

1. '커뮤니티' 서비스는 사용자들이 소통할 수 있는 공간으로서 관리자에 의해 제어됩니다.
2. 관리자는 '관리자 페이지' 에서 '커뮤니티' 서비스를 추가/삭제 할 수 있습니다.
3. '커뮤니티' 서비스 추가 시 새로운 성격의 게시판이 생성되며 추가된 게시판에서 회원들은 소통할 수 있습니다.
4. '커뮤니티' 서비스 삭제 시 해당 서비스에 종속된 게시글 모두 삭제됩니다.

MODEL	<div><div>1. Service</div><div>① 커뮤니티 목록</div><pre>@Override public ListDTO&lt;BoardDTO&gt; getComCategory(int cno,int pageNo) {     int totalCount=adminDAO.getBoardCountPerCategory(cno);     PagingBean pb=null;     if(pageNo==1)         pb=new PagingBean(10,4,totalCount);     else         pb=new PagingBean(pageNo, 10, 4, totalCount);     return new ListDTO&lt;Dto&gt;(communityDAO.getComCategory(pb,cno), pb); }</pre><div>② 커뮤니티 추가</div><pre>@Override public BoardDTO registerCommunityInfo(BoardDTO boardDTO) {     return communityDAO.registerCommunityInfo(boardDTO); }</pre><div>③ 커뮤니티 삭제</div><pre>@Override public void removeCommunityInfo(int boardNo) {     communityDAO.removeCommunityInfo(boardNo); }</pre><div>2. DAO</div><div>① 커뮤니티 목록</div><pre>@Override public List&lt;BoardDTO&gt; getCommunityListPerCategory(PagingBean     pagingBean,int categoryNo) {     HashMap&lt;String, Object&gt; p=new HashMap&lt;String, Object&gt;();     p.put("categoryNo", categoryNo);     p.put("startRowNumber", pagingBean.getStartRowNumber());     p.put("endRowNumber", pagingBean.getEndRowNumber());     return t.selectList("community.communityListPerCategory",p); }</pre><div>② 커뮤니티 추가</div><pre>@Override public void registerAddCommunityType(String categoryName) {     template.insert("admin.addCommunityType",categoryName); }</pre></div>
-------	--

	<p>③ 커뮤니티 삭제</p> <p>a. 커뮤니티 타입 삭제</p> <pre>@Override public void removeCommunityType(int categoryNo) {     template.delete("admin.removeCommunityType",categoryNo); }</pre> <p>b. 커뮤니티 종속 게시물 삭제</p> <pre>@Override public void removeCommunityTypeBoard(int categoryNo) {     template.delete("admin.removeCommunityTypeBoard",categoryNo); }</pre>
VIEW	<p>1. 해당 서비스에 접근하기위한 경로 생성</p> <pre>&lt;c:forEach items="\${categoryType}" var="ct"&gt;     &lt;a href="community.do?categoryNo=\${ct.categoryNo}&amp;categoryName=         \${ct.Name}&amp;pageNo=1"&gt;\${ct.Name}     &lt;/a&gt; &lt;/c:forEach&gt;</pre>
CONTROLLER	<p>1. 커뮤니티 목록</p> <pre>@RequestMapping("community.do") public String communityListRequest(int no,String name,Model m,int pno) {     m.addAttribute("cn",name);     m.addAttribute("cmulist",communityService.getComCategory(no,pno));     model.addAttribute("categoryType",communityService.getCatelist());     return "community.tiles"; }</pre> <p>2. 커뮤니티 추가</p> <pre>@RequestMapping("addcommunitytype.do") public String addCommunityTypeRequest(Model model,String categoryName) {     adminService.registerAddCommunityType(categoryName);     return "redirect:communitycategory.do"; }</pre> <p>3. 커뮤니티 삭제</p> <pre>@RequestMapping("deleteCommunityType.do") public String removeCommunityTypeRequest(int no){     adminService.removeCommunityType(no);     return "redirect:communitycategory.do"; }</pre>

## 2)모임 참여자 관리

### Service Content :

1. 모임 개설자는 자신의 모임에 참여한 참가자들을 확인/관리할 수 있습니다.
2. 참가한 관리자들의 명단을 확인하고 강퇴 할 수 있습니다.

MODEL	<pre> 1. service @Override @Transactional public List&lt;MemberDTO&gt; getMyMeetingAttender(int boardNo) {     return dao.getMyMeetingAttender(boardNo); } @Override public void removeAttendByFounder(String id, int boardNo) {     dao.removeAttendByFounder(id, boardNo); }  2. dao @Override public List&lt;MemberDTO&gt; getMyMeetingAttender(int boardNo) {     return template.selectList("meeting.myMeetingAttender", boardNo); } @Override public void removeAttendByFounder(String id, int boardNo) {     HashMap&lt;String, Object&gt; param=new HashMap&lt;String, Object&gt;();     param.put("id", id);     param.put("boardNo", boardNo);     template.delete("meeting.removeAttendByFounder", param); } </pre>
CONTROLLER	<pre> 1. 참여자 리스트 @RequestMapping("meetingAttenderList.do") public String meetingAttenderListRequest(int boardNo,Model model) {     model.addAttribute("boardNo", boardNo);     model.addAttribute("atlist",service.getMyMeetingAttender(boardNo))     return "popup/popup.tiles"; }  2. 참여자 제거 @RequestMapping(value="exceptAttend.do",method=RequestMethod.POST) @ResponseBody public String removeAttendByFounder(String id, String boardNo) {     service.removeAttendByFounder(id, Integer.parseInt(boardNo));     return id; } </pre>

## OPEN API

### 1)SMART EDITOR

#### Service Content :

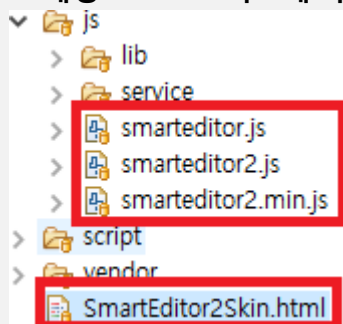
SmartEditor Basic 2.0은 네이버에서 제공하는 텍스트 편집 라이브러리로서, JavaScript로 구현된 웹 기반의 WYSIWYG 편집기입니다. 글꼴, 글자 크기, 줄 간격 등을 자유롭게 설정할 수 있으며, 단어 찾기/바꾸기와 같은 편리한 기능을 제공합니다.

#### API :

##### 1. API URL

[http://naver.github.io/smarteditor2/user\\_guide/](http://naver.github.io/smarteditor2/user_guide/)

##### 2. 해당API 프로젝트에 추가



##### 3. Script

```
<script type="text/javascript" src="resources/js/service/HuskyEZCreator.js" charset="utf-8" />
```

##### 4. TextArea 선언

```
<textarea name="content" id="ir1" rows="10" cols="100" style="width:766px; height:412px; display:none;"></textarea>
```

##### 5. Script code(submit)

```
oEditors.getById["ir1"].exec("UPDATE_CONTENTS_FIELD", []);
```

## 2)Daum Maps

### Service Content :

Daum 지도 API는 웹사이트와 모바일 어플리케이션에서 지도를 이용한 서비스를 제작할 수 있도록 다양한 기능을 제공하고 있습니다. Daum 지도 API를 이용하여 제작된 서비스들을 둘러보세요. 웹사이트와 모바일 어플리케이션에서 다양하게 활용되고 있습니다.

### 서비스 이용순서 :

1. 카카오 개발자사이트 (<https://developers.kakao.com>) 접속
2. 개발자 등록 및 앱 생성
3. 웹 플랫폼 추가: 앱 선택 - [설정] - [일반] - [플랫폼 추가] - 웹 선택 후 추가
4. 사이트 도메인 등록: [웹] 플랫폼을 선택하고, [사이트 도메인] 을 등록합니다. (예: <http://localhost:8080>)
5. 페이지 상단의 [JavaScript 키]를 지도 API의 appkey로 사용합니다.
6. 앱을 실행합니다.

### API :

#### 1. 지도영역 Tag

```
<div id="map" style="width:500px;height:400px;"></div>
```

#### 2. Script

```
<script type="text/javascript" src="//dapi.kakao.com/v2/maps/sdk.js?appkey=발급받은 APP KEY"></script>
```

#### 3. Script Code

```
var container = document.getElementById('map'); // 지도를 담을 영역의 DOM 레퍼런스
var options = { // 지도를 생성할 때 필요한 기본 옵션
    center: new daum.maps.LatLng(33.450701, 126.570667), // 지도의 중심좌표.
    level: 3 // 지도의 레벨(확대, 축소 정도)
};

var map = new daum.maps.Map(container, options); // 지도 생성 및 객체 리턴
```



### 3)Kakao Pay

#### Service Content :

아임포트는 KG이니시스, NHN KCP, 나이스정보통신, LG유플러스, JNet, 다날휴대폰결제, 카카오페이, Paypal 등 국내외 주요 PG사와의 연동을 지원합니다. Javascript 기반의 웹 SDK. Javascript API만 호출하면 PG 결제창 연동까지 한 번에 해결합니다. 파라미터 입력만으로 결제정보를 변경하는 직관적인 제어가 가능합니다.

#### API :

##### 1. Script

```
<script  
type="text/javascript"src=https://service.iamport.kr/js/iamport.payment-1.1.5.js />
```

##### 2. Script Code

```
IMP.init('imp99239962');  
$("#accountPay").click(function () {  
    var pay = $("#pay").val();  
    IMP.request_pay({  
        pg : 'html5_inicis',  
        pay_method : 'card',  
        merchant_uid : 'merchant_' + new Date().getTime(),  
        name : '주문명:결제테스트',  
        amount : pay,  
        buyer_email : 'iamport@siot.do',  
        buyer_name : '구매자이름',  
        buyer_tel : '010-1234-5678',  
        buyer_addr : '서울특별시 강남구 삼성동',  
        buyer_postcode : '123-456'  
    }, function (rsp) {  
        if ( rsp.success ) {  
            var msg = '결제가 완료되었습니다.';  
            msg += '고유ID : ' + rsp.imp_uid;  
            msg += '상점 거래ID : ' + rsp.merchant_uid;  
            msg += '결제 금액 : ' + rsp.paid_amount;  
            msg += '카드 승인번호 : ' + rsp.apply_num;  
            $("#payform").submit();  
        } else {  
            var msg = '결제에 실패하였습니다.';  
            msg += '에러내용 : ' + rsp.error_msg;  
        }  
        alert(msg);  
    });  
});
```

## 4)Kakao Login

### Service Content :

카카오계정을 통한 간편 로그인을 제공합니다. 이외에도 사용자들의 개별 정보를 손쉽게 관리해주는 기능을 포함합니다. 카카오 플랫폼 서비스에서 제공하는 기능 중 로그인이 필요한 기능을 사용하기 위해서는 반드시 선행되어야 할 내용들을 다룹니다.

### 서비스 이용 순서 :

- 1.새로운 앱을 만듭니다. 앱 만들기
- 2.앱의 설정 - 일반 에서 플랫폼 추가를 선택합니다.
- 3.웹 플랫폼을 선택하고, 사이트 도메인 을 등록합니다. (예: http://localhost:8080 )
- 4.페이지 상단의 JavaScript 키 항목을 외우거나, 복사해 둡니다.

### API :

#### 1. Script

```
<script src="//developers.kakao.com/sdk/js/kakao.min.js"></script>
```

#### 2. 버튼 생성 영역

```
<a id="kakao-login-btn"></a>
<a href="http://developers.kakao.com/logout"></a>
```

#### 3. Script code

```
<script type='text/javascript'>
  Kakao.init('YOUR APP KEY');

  Kakao.Auth.createLoginButton({

    container: '#kakao-login-btn',

    success: function(authObj){

      alert(JSON.stringify(authObj));

    },fail: function(err) {

      alert(JSON.stringify(err));

    }

  });
</script>
```

## 6 평가

### 6.1 Best Practice

아이템 구성
<p><b>“반려견들을 위한 커뮤니티”</b></p> <p>반려인구 천만 시대가 도래했음에도 반려견을 대하는 우리의 태도 혹은 인식은 아직 미성숙하다. 이에 ‘함께하시개’는 각종 모임, 교육, 세미나, 물품 나눔 등을 통해 견주들이 반려견을 키우는데 필요한 지식과 정보를 공유할 수 있도록 한다. 그리고 반려견들의 사회성을 신장시킬 수 있도록 산책 모임, 산책 도우미 등 친목 교류의 장을 마련한다.</p>
솔루션
<p><b>“다양한 API를 활용한 서비스”</b></p> <p>회원가입 없이 간편한 로그인을 위한 카카오로그인 API를 사용하였다. 그리고 함께하시개의 주요 기능 중 하나인 모임개설을 위해 다음 지도 API를 사용하여 모임 장소를 좀 더 정확하고 편리하게 표시할 수 있게 하였다. 또한 같이가치 모금함 기능을 위해 카카오 페이 API를 사용했다.</p>
프로젝트 관리
<p><b>“칸반과 마일스톤”</b></p> <p>프로젝트를 보다 효과적으로 관리하기 위한 도구로는 칸반 보드를 사용했다. 칸반 보드를 활용하여 팀 업무를 시각화하고, 워크플로우를 표준화하며, 프로젝트 진행 중에 발생하는 장애물 및 종속성을 빠르게 식별하고 해결할 수 있었다. 그리고 마일스톤을 통해 체계적인 일정관리를 했다. 마일스톤에 명시된 작업 내용을 매일 확인하여 현재의 진행 상태를 파악할 수 있었다. 또한 각 단계의 마무리 시점을 표시함으로써 복잡한 일정을 효율적으로 관리하였다.</p>

## 6.2 Lesson Learned

프로젝트 관리
<p><b>“매일 매일 회의를 하며 진행상황 점검”</b></p> <p>프로젝트를 진행함에 있어 우리 팀원들이 가장 중요하게 생각한 것은 ‘원활한 의사소통’ 이었다. 그래서 프로젝트 시작 단계부터 종료 시점까지 매일 회의를 통해 각자의 의견을 공유하고, 이슈 사항을 해결하는 시간을 가졌다. 그리고 이런 회의 내용을 문서화하여 시간이 흐른 뒤에도 당시의 회의 내용과 이슈 사항을 파악할 수 있도록 하였다. 이렇게 매일 회의를 함으로써 진행상황을 보다 세밀하게 관리할 수 있었기에 프로젝트를 무사히 마무리 할 수 있었다.</p>
구현과정
<p><b>“유지보수성을 높이기 위해 프레임워크 활용”</b></p> <p>Spring Framework를 기반으로 진행한 이번 프로젝트를 통해 단순히 기능을 구현하는 정도에서 끝나는 것이 아니라, 후에 유지 보수를 위한 코드 구현에 대해 많이 생각하는 계기가 되었다. 인터페이스와 상속을 통해 계층구조를 형성하여 추상화의 깊이를 높이고, IO C의 DI를 통해 객체를 주입 받는 등 유지보수성을 생각하는 개발에 대해서도 생각할 수 있었다.</p>
협업
<p><b>“협업을 위한 역할 분담”</b></p> <p>프로젝트 기능 구현에서의 역할 뿐만 아니라, ‘밥짱(오늘의 식단을 정하는 사람)’, ‘포토그래퍼’ 등의 역할을 분담하여 팀의 분위기를 환기시켰다. 또한 팀구호를 만들어 화이팅이 필요한 순간마다 구호를 외쳐 기운을 북돋울 수 있었다. 이렇게 각자 맡은 기능 구현과 팀 내 역할을 수행해 나가면서 서로를 배려하는 팀 문화를 경험할 수 있었다.</p>