

SNU Moyeo : “혼자 하면 힘든, 같이 하면 쉬운”

Design and Planning

Members of group 2

최덕경 (컴퓨터공학부 2016-10399)

김동욱 (컴퓨터공학부 2016-16712)

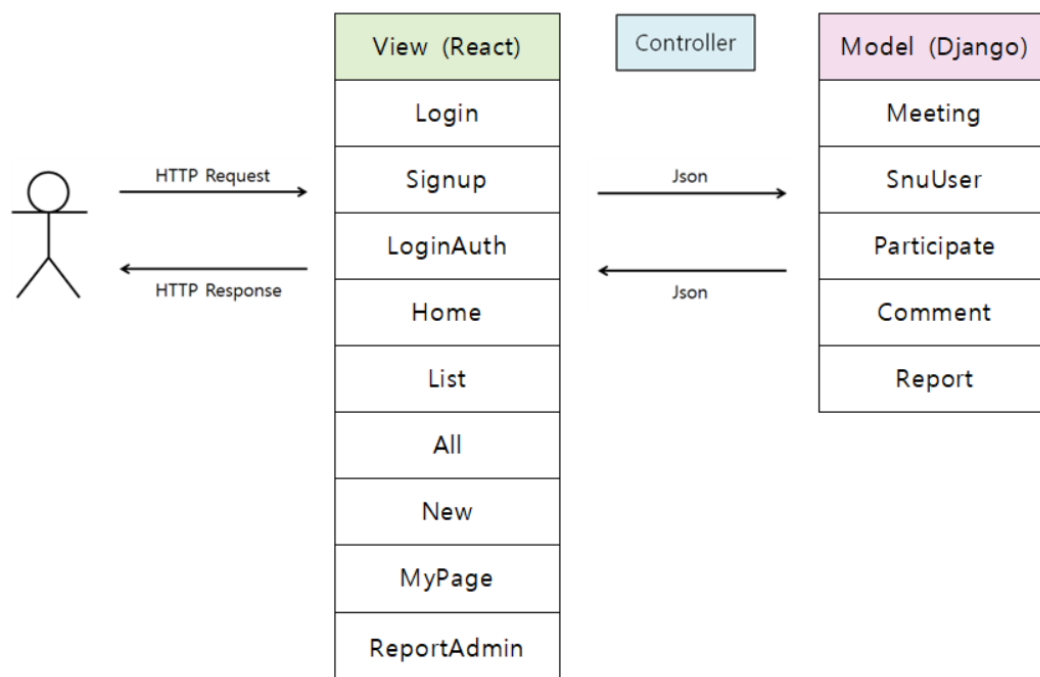
이종민 (컴퓨터공학부 2016-15160)

정재훈 (컴퓨터공학부 2016-11464)

System Architecture

1) MVC (Model-View-Controller)

우리 웹 서비스의 MVC 모델은 다음과 같다. 프론트 엔드의 뷰는 총 9개 존재하며, 백 엔드의 모델은 총 5개 존재한다. 컨트롤러 부분은 뒷부분에서 추후 더 논의할 것이다.

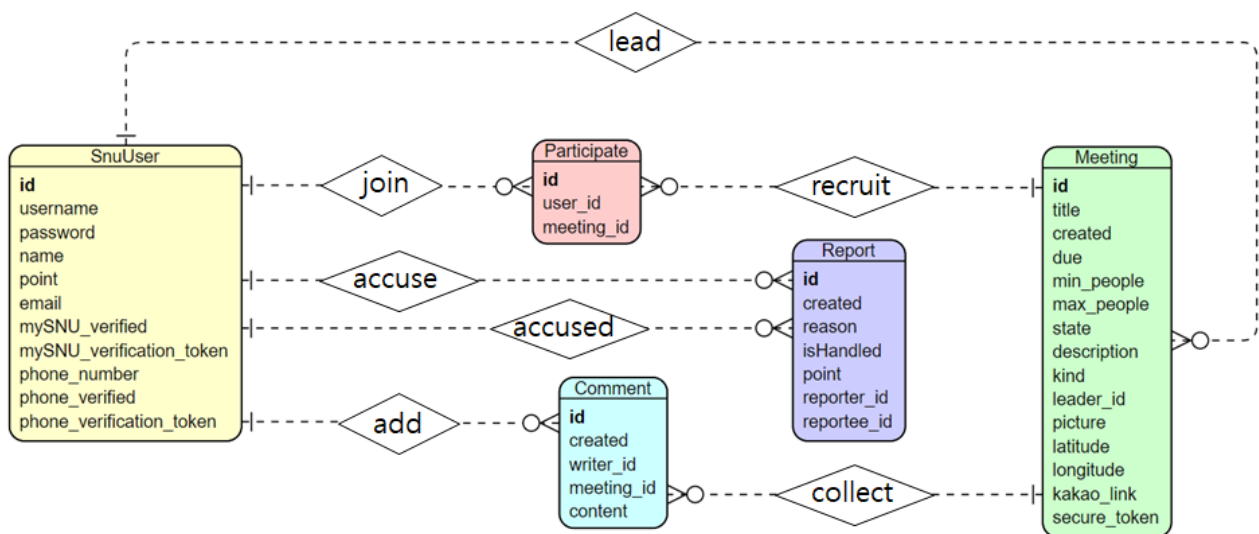


2) Model

Entity Relationship Diagram (ERD)

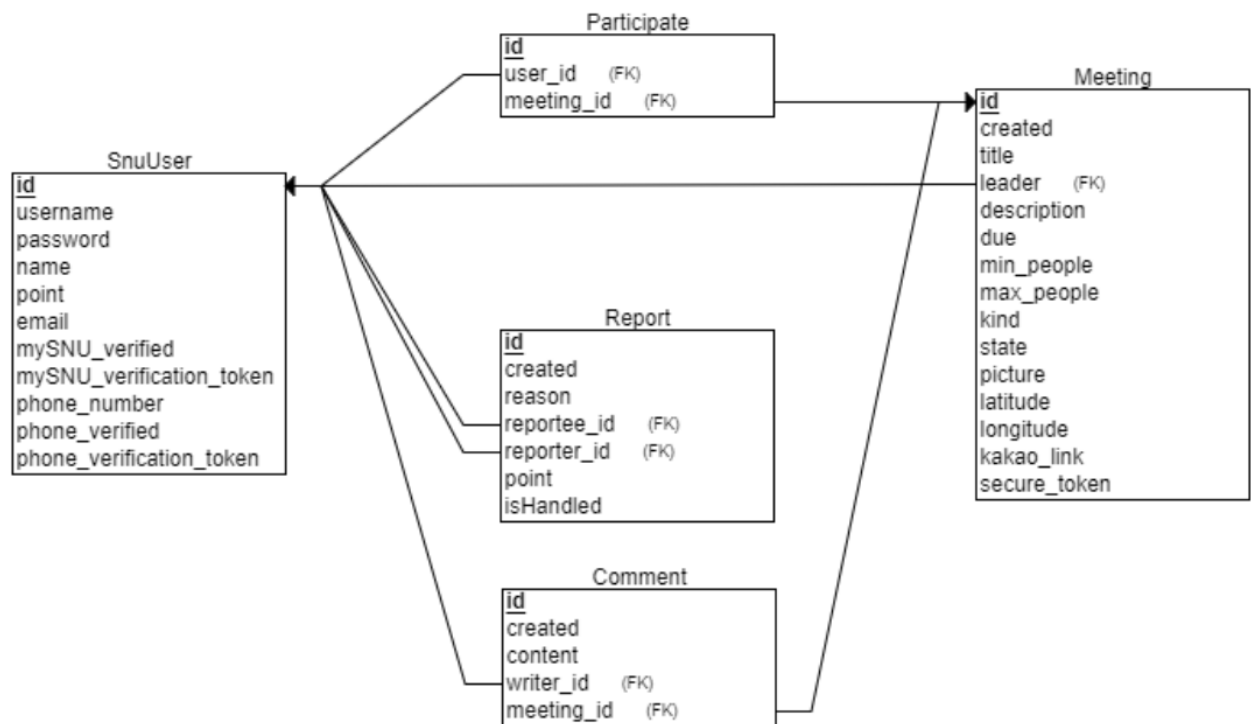
MVC 구조 중 모델을 디자인하기 위해 ER 다이어그램을 도식화하였다. 사각형 각각은 하나의 개체 타입을 나타내고, 마름모 각각은 하나의 관계 타입을 나타낸다. 관계 타입은 개체 타입들 간의 관계를 알려준다. 또한 각 개체 타입이 가지는 속성들은 해당 사각형 안에 나열되어 있다(일반적으로는 타원으로 표현하여 개체 타입과 선으로 연결하지만 속성이 많은 관계로 간소화를 위해 이 방법을 채택하였다). 속성들 중 밑줄이 그어져 있는 것은 해당 개체 타입의 기본키이다.

- 한 유저는 여러 모임을 주최(lead)할 수 있고, 한 모임은 오로지 한 유저에 의해서만 주최될 수 있다.
- 한 유저는 여러 모임에 참가(join)할 수 있고, 한 모임은 여러 유저를 모집(recruit)할 수 있다. (→ 교차 엔티티(Association Entity) 활용)
- 한 유저는 다른 유저들에 대해 여러 번 신고할 수 있고(accuse), 한 유저는 다른 유저들에 의해 여러 번 신고받을 수 있으며(accused), 하나의 신고 접수는 신고자에 해당하는 한 유저와 신고 대상에 해당하는 한 유저를 가질 수 있다.
- 한 유저는 여러 개의 댓글을 달 수 있으며(add), 하나의 댓글은 댓글 작성자로서 오로지 한 유저만을 가질 수 있다.
- 한 모임은 여러 개의 댓글을 가질 수 있으며(collect), 한 댓글은 오로지 한 모임에만 종속되어 있다.



Relation Schema Diagram based on ERD

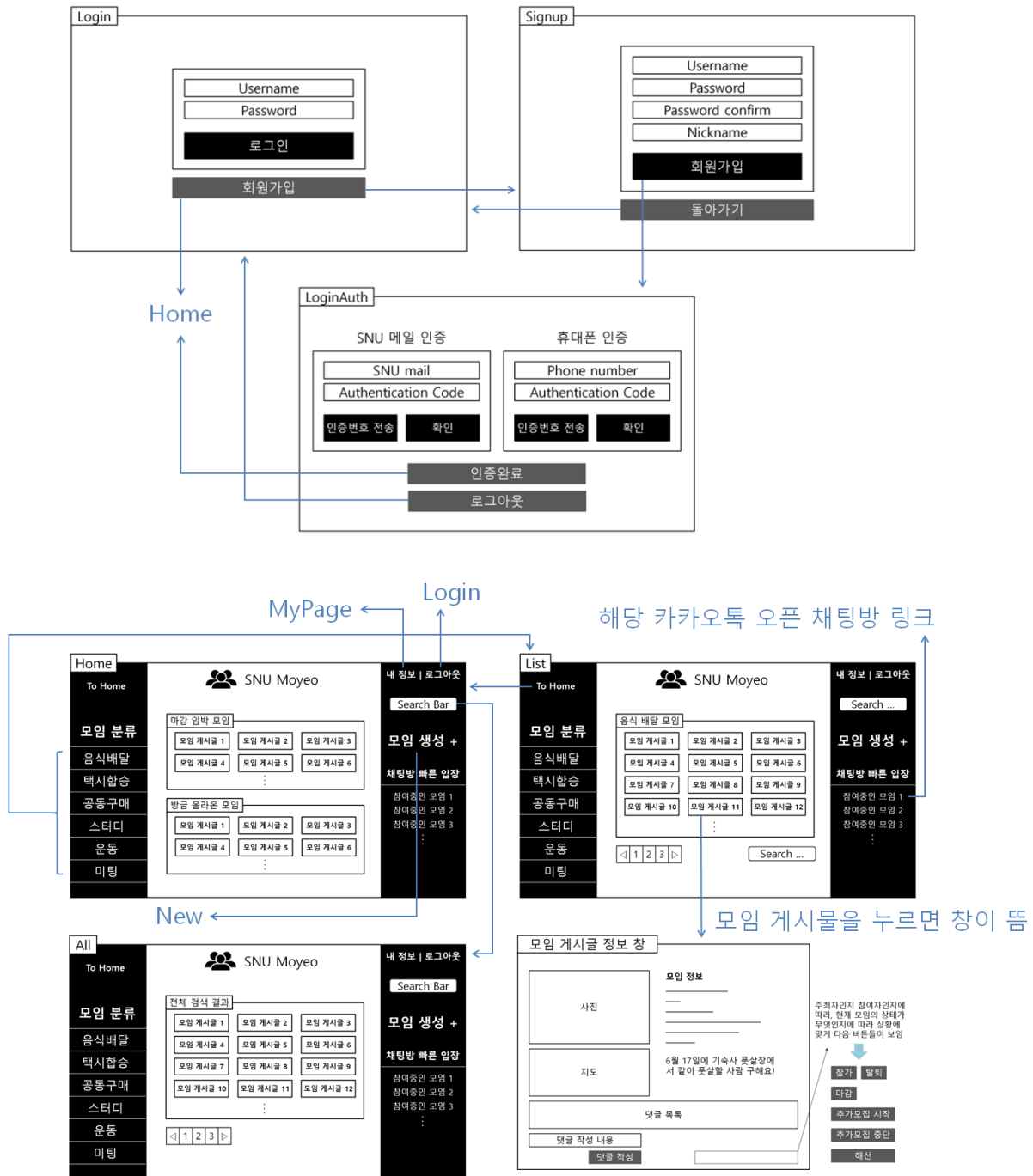
위에서 도식화한 ER 다이어그램을 기반으로 개체들 간 관계를 더욱 명확하게 나타내는 Relation Schema를 도식화하였다. 이는 백 엔드에서 장고 모델을 디자인할 때 기반이 되는 구조이다. (FK)는 외래키를 나타내는 기호이며, 각 화살표는 해당 외래키가 가리키는 모델을 나타낸다.



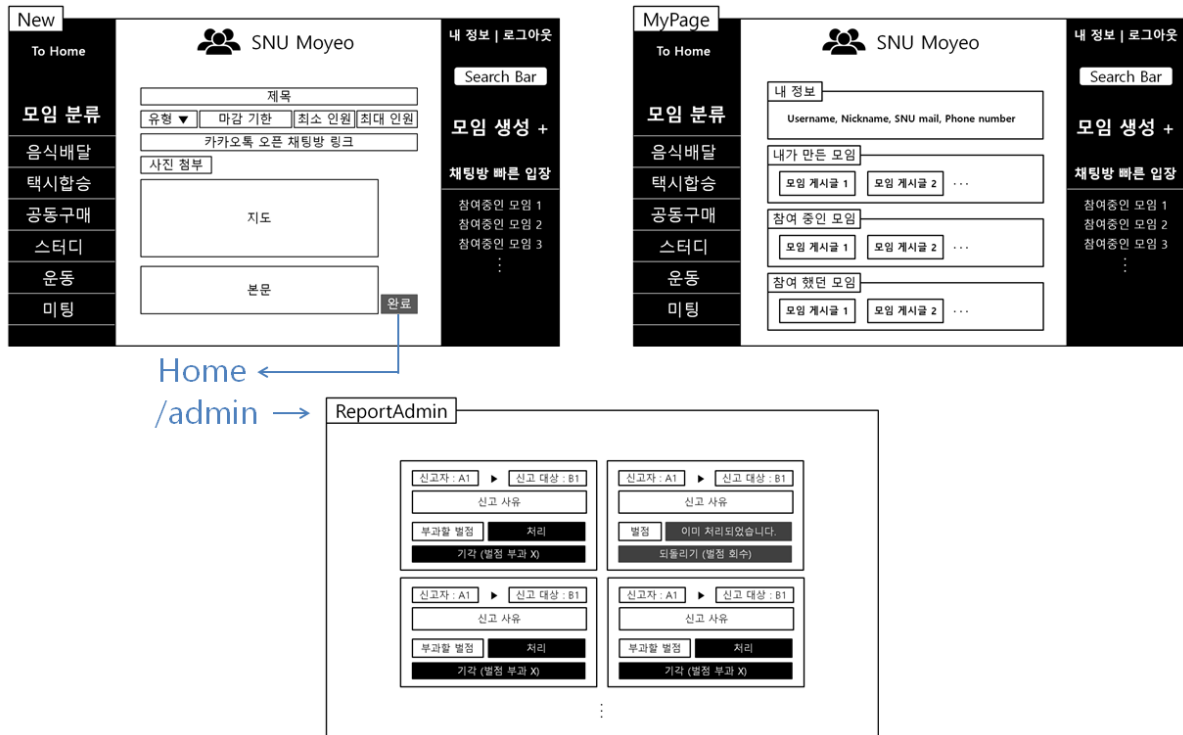
3) View

User Interface (UI)

MVC 구조 중 뷰를 디자인하기 위해 클라이언트에게 보여줄 유저 인터페이스를 디자인하였다. 프론트 엔드의 뷰는 총 9개(Login, Signup, LoginAuth, Home, All, List, New, MyPage, ReportAdmin)로 이루어져 있으며 각각의 모습과 그들 간의 관계는 다음과 같다.



※ 중복되는 화살표는 하나만 표시하였음



Functionality and Requirement

각 페이지의 기능과 요구 사항은 아래에 명시되어 있다. 기본적으로는 페이지 별로 나누어서 유저가 접할 수 있는 기능들을 정리하였다. 단, 자주 사용하는 공통 기능들은 따로 떼어서 설명하였다. 예를 들어 모임 게시물을 눌렀을 때 뜨는 창에서 취할 수 있는 행동의 경우 여러 페이지에서 공통으로 경험할 수 있는 상황이기 때문에 페이지 별로 설명하지 않고 하나로 묶어서 설명하였다.

1) In Login Page ('/login')

- 회원가입 : [회원가입] 버튼을 누르면 회원가입 페이지로 이동한다.
- 로그인 : 인증된 유저가 아이디와 패스워드를 올바르게 입력하여 로그인하면 홈 페이지로 이동한다. 만약 인증되지 않은 유저가 로그인하면 인증 페이지로 이동한다.

2) In Signup Page ('/signup')

- 회원가입 : 아이디, 패스워드, 이름을 올바르게 입력하고 보안문자 인증까지 완료하면 성공적으로 회원가입이 이뤄진다.
- 돌아가기 : [돌아가기] 버튼을 누르면 로그인 페이지로 되돌아간다.

3) In LoginAuth Page ('/auth')

- MySNU 메일 인증 : MySNU 메일을 입력하고 인증번호를 요청한 뒤 인증번호를 올바르게 입력하면 MySNU 인증이 성공적으로 이뤄진다.
- 휴대폰 인증 : 휴대폰 번호를 입력하고 인증번호를 요청한 뒤 인증번호를 올바르게 입력하면 휴대폰 인증이 성공적으로 이뤄진다.
- 인증완료 : MySNU 메일 인증과 휴대폰 인증을 모두 완료한 뒤 [인증완료] 버튼을 누르면 홈 페이지로 이동한다.
- 로그아웃 : [로그아웃] 버튼을 누르면 로그인 페이지로 되돌아간다.

4) In Home Page ('/')

- 마감 임박 모임 : 모든 모임 게시물들을 마감 날짜가 가까운 순서대로 정렬하여 가장 마감이 가까운 6개의 모임 게시물만 보여준다.
- 딱끈따끈 방금 올라온 모임 : 모든 모임 게시물들을 최근에 게시된 순서대로 정렬하여 가장 최근에 게시된 6개의 모임 게시물만 보여준다.
- 모임 게시물을 누르면 해당 모임의 정보를 살펴볼 수 있다.

5) In List Page ('/list:n')

- 특정 유형(n = 유형 번호)의 모임 게시물들을 최근에 게시된 순서대로 정렬하여 보여준다.
- 검색 : 하단 검색창에 검색어를 입력함으로써 원하는 모임 게시물을 검색할 수 있다.
- 네이버 쇼핑 검색 : 만약 공동구매 유형의 모임 게시판이라면 검색창 바로 아래에 있는 또 다른 검색창을 이용하여 네이버 쇼핑에서 검색을 해볼 수 있다. 검색 시 유사도 기준으로 상위 10개의 검색 결과가 팝업으로 뜨게 된다.

6) In All Page ('/all')

- 우측 사이드바에 존재하는 전체 검색창에 검색어를 입력하여 검색했을 때 검색 결과를 볼 수 있는 페이지이다. 검색어를 포함하는 모든 유형의 모임 게시물들을 보여준다.

7) In List, or All Page ('/list:n', or '/all')

- 페이지네이션 : 모임 게시물들이 페이지 별로 나뉘어 있기 때문에 페이지 번호를 누르거나 화살표 버튼을 누르면 페이지를 이동할 수 있다.

8) New page ('/new')

- 모임 생성 : 제목, 유형, 마감 날짜, 최소/최대 인원, 내용을 입력하고 선택적으로 사진을 첨부하거나 위치를 다음 맵으로 지정하거나 카카오톡 오픈 채팅방 링크를 작성한 뒤 [완료] 버튼을 누

르면 모임 게시물이 게시되면서 홈 페이지로 이동한다.

- 네이버 쇼핑 검색 : 만약 유형으로 공동구매를 선택하면 네이버 쇼핑 검색창이 나타난다. 여기에 검색어를 입력하면 네이버 쇼핑에서 검색을 해볼 수 있다. 검색 시 유사도 기준으로 상위 10개의 검색 결과가 팝업으로 뜨게 된다.

9) In MyPage Page ('/mypage')

- 내 정보 확인 : 나의 아이디, 이름, MySNU 메일 주소, 휴대폰 번호를 확인할 수 있다.
- 내가 만든 모임 : 내가 만든 모임들이 보이고, 모임 게시물을 누르면 모임 정보를 볼 수 있다.
- 내가 참여중인 모임 : 내가 참여중인 모임들이 보이고, 같은 방식으로 모임 정보를 볼 수 있다.
- 내가 참여했던 모임 : 내가 참여했던 모임들이 보이고, 같은 방식으로 모임 정보를 볼 수 있다.

10) ReportAdmin page ('/admin')

- 신고 처리 : 부여할 벌점 점수를 입력하고 처리 버튼을 누르면 신고 대상에게 벌점을 준다.
- 신고 기각 : 기각 버튼을 누르면 해당 신고는 기각된다.
- 신고 되돌리기 : 이미 처리된 신고의 [되돌리기] 버튼을 누르면 벌점을 회수한다.

11) In Home, List, All, New, or MyPage Page ('/', '/list/:n', '/all', '/new', or '/mypage')

- [To Home] 버튼을 누르면 홈 페이지로 되돌아간다.
- [내 정보] 버튼을 누르면 내 정보 페이지로 이동한다.
- [로그아웃] 버튼을 누르면 로그인 페이지로 되돌아간다.
- 좌측 사이드바에 있는 모임 유형을 클릭하면 해당 유형의 모임 게시물 목록을 볼 수 있다.
- 우측 사이드바에 있는 [새 모임 만들기 +] 버튼을 클릭하면 모임 생성 페이지로 이동한다.
- 우측 사이드바에 있는 전체 검색창에 검색어를 입력하여 검색하면 검색어를 포함하는 모든 유형의 모임 게시물들을 볼 수 있다.
- 우측 사이드바에는 내가 참여중인 모임의 카톡 오픈 채팅방 링크가 표시되어 있다.

12) When Clicking a Meeting Post

- 다른 사람이 만든 모임 게시물 : 제목, 본문, 리더, 마감 날짜, 지도, 사진, 댓글 목록 등의 기본 모임 정보를 확인할 수 있다. 내가 참여 중이라면 카톡 오픈 채팅방 링크도 확인할 수 있고 [탈퇴] 버튼을 클릭하여 탈퇴가 가능하다. 아니면 [참가] 버튼을 클릭하여 참가가 가능하다.
- 내가 만든 모임 게시물 : 마찬가지로 기본 모임 정보를 확인할 수 있고 [수정] 버튼을 클릭하여 모임 게시물 수정이 가능하며 [해산] 버튼을 눌러서 해산이 가능하다. 또한 참여자들 정보를 확인할 가능하며 이를 담은 엑셀 파일도 다운 가능하다. 이때 상태가 [모집 중]이라면 [마감] 버튼이, [마감]이라면 [추가모집 시작] 버튼이, [추가모집 중] 이라면 [추가모집 중단] 버튼이 보인다.

13) In Popup Window Showing Meeting info

- 댓글 작성 : 댓글을 작성하고 [댓글 작성] 버튼을 누르면 댓글이 달린다.
- 댓글 수정 : 내가 작성한 댓글에 있는 [수정] 버튼을 눌러서 댓글 수정이 가능하다.
- 댓글 삭제 : 내가 작성한 댓글에 있는 [삭제] 버튼을 눌러서 댓글 삭제가 가능하다.
- 모임 참가 (내가 참여하지 않은 모임일 때) : [참가] 버튼을 눌러서 참가가 가능하다.
- 모임 탈퇴 (내가 참여중인 모임일 때) : [탈퇴] 버튼을 눌러서 탈퇴가 가능하다.
- 모임 마감 (내가 만든 모임일 때) : [모집 중] 상태이고 최소인원이 충족됐으면 [마감] 버튼을 눌러서 모집 마감이 가능하다. 그러면 더 이상 다른 유저에게 해당 모임 게시물이 보이지 않는다.
- 모임 추가모집 시작 (내가 만든 모임일 때) : [마감] 상태이면 [추가모집 시작] 버튼을 눌러서 추가모집 시작이 가능하다. 그러면 다시 다른 유저에게 해당 모임 게시물이 보이게 된다.
- 모임 추가모집 중단 (내가 만든 모임일 때) : [추가모집 중] 상태이면 [추가모집 중단] 버튼을 눌러서 추가모집 중단이 가능하다. 그러면 더 이상 다른 유저에게 해당 모임 게시물이 보이지 않는다.
- 모임 해산 (내가 만든 모임일 때) : [해산] 버튼을 눌러서 모임 해산이 가능하다.
- 모임 수정 (내가 만든 모임일 때) : [수정] 버튼을 눌러서 모임 수정이 가능하다.
- 주최자 정보 확인 및 신고 : 주최자의 이름을 클릭하고 [정보보기]를 클릭하면 정보를 확인할 수 있고 [신고하기]를 클릭하면 신고 사유를 적고 신고가 가능하다.
- 참여 멤버 정보 확인 및 신고 (내가 만든 모임일 때) : 참여 인원을 누르면 참여자들의 정보가 뜬다. 그 중에 한 참여자의 이름을 클릭하고 [정보보기]를 클릭하면 정보를 확인할 수 있고 [신고하기]를 클릭하면 신고 사유를 적고 신고가 가능하다. 또한 [참여 멤버 정보 다운] 버튼을 클릭하면 참여자들의 정보를 담은 엑셀 파일을 다운받을 수도 있다.
- 카톡 오픈 채팅방 링크 확인 (내가 참가한 모임일 때) : 오픈 채팅방 링크를 확인 가능하다.

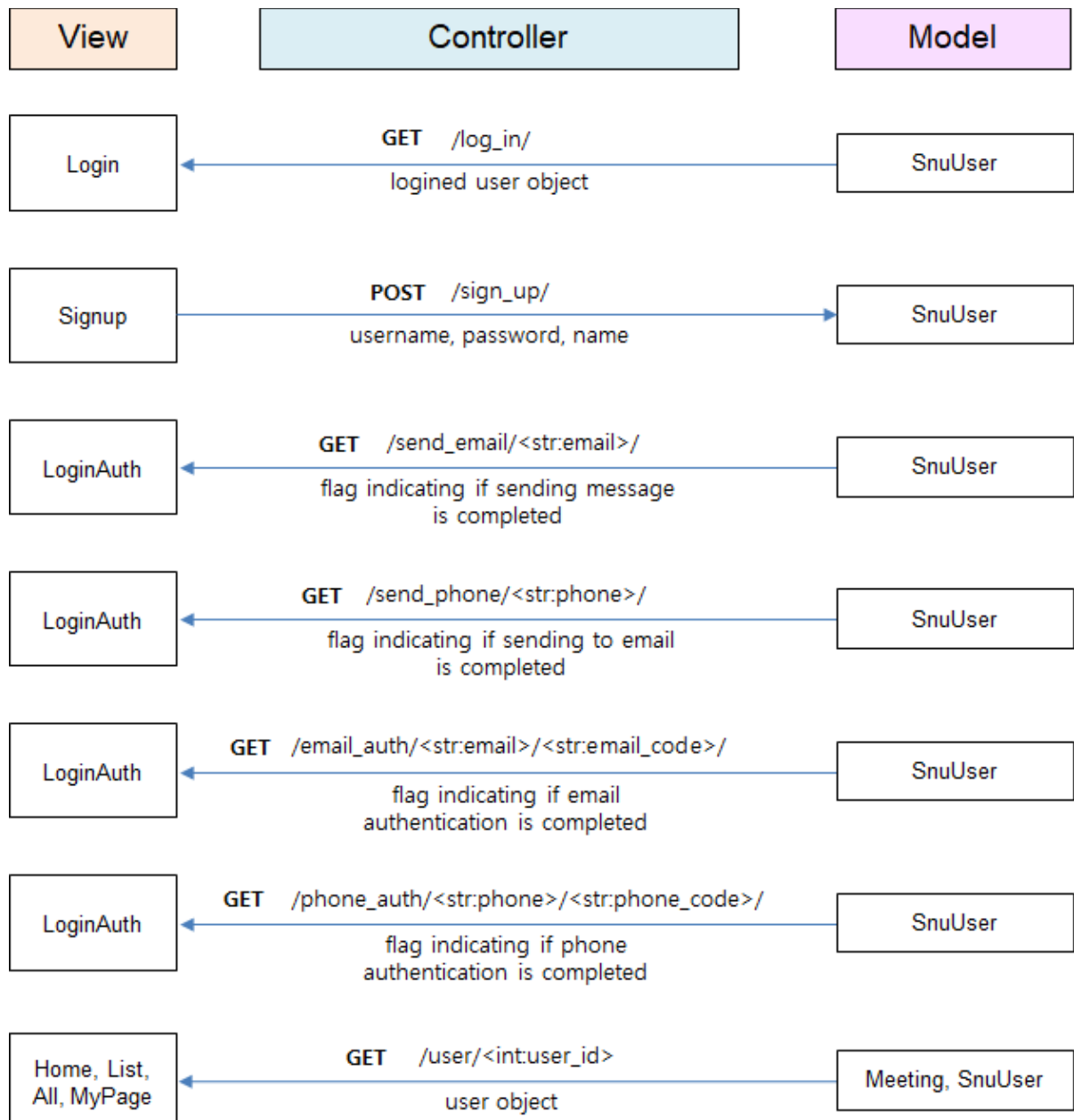
14) Redirection

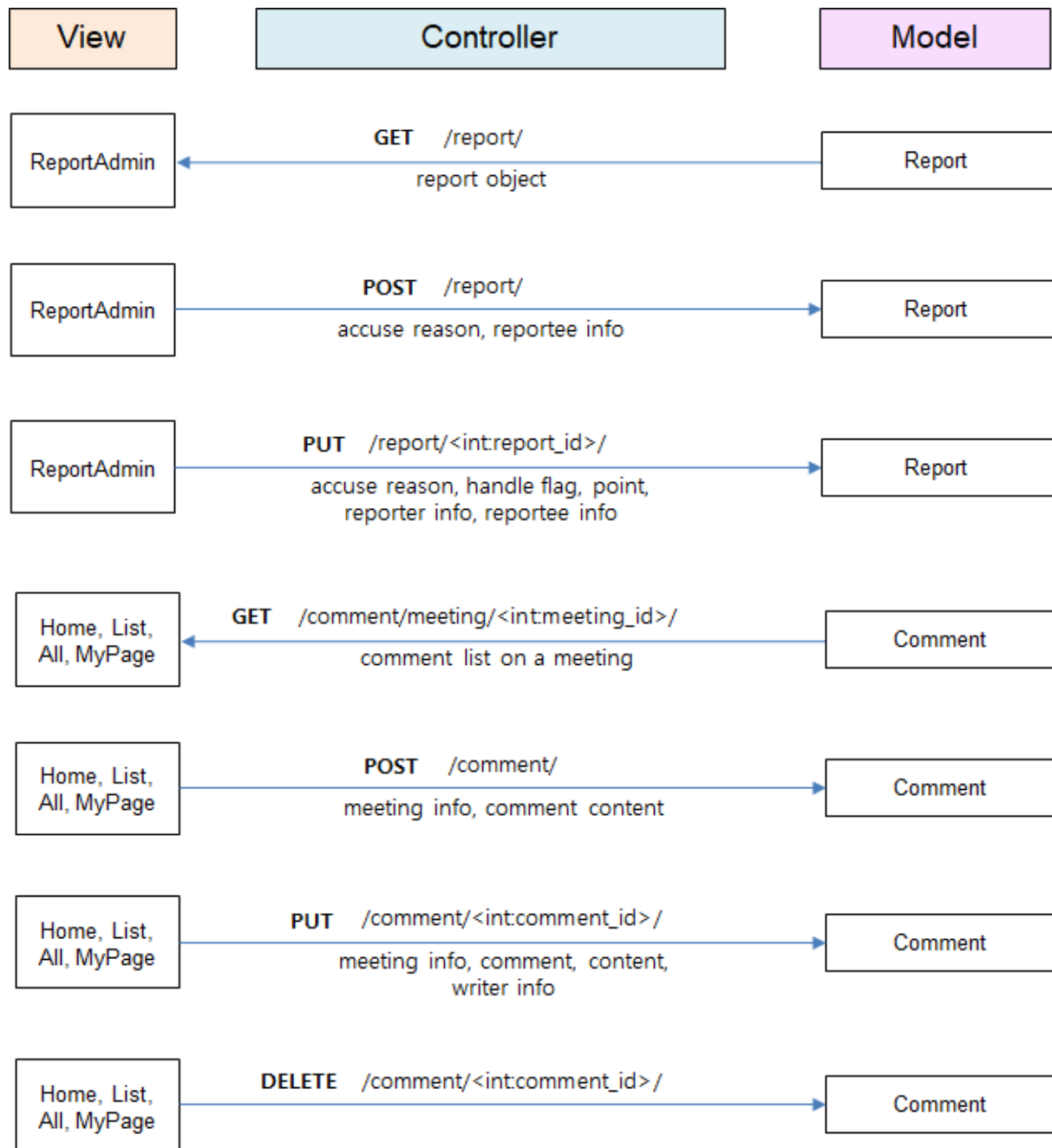
- 로그인을 안 했을 때 : Home, MyPage, New, List, All, LoginAuth, ReportAdmin 페이지 중 하나에 URL 을 타고 접속을 시도하면 로그인 페이지로 리다이렉트된다.
- 미인증 유저일 때 : Login, Signup, Home, MyPage, New, List, All 페이지 중 하나에 URL 을 타고 접속을 시도하면 인증 페이지로 리다이렉트된다.
- 로그인한 상태일 때 : Login, Signup 페이지 중 하나에 URL 을 타고 접속을 시도하면 홈 페이지로 리다이렉트된다.
- 관리자가 아닐 때 : ReportAdmin 페이지에 URL 을 타고 접속을 시도하면 홈 페이지로 리다이렉트된다.
- 별점이 10 점 이상인 유저일 때 : Login, Signup 페이지를 제외한 페이지에 URL 을 타고 접속을 시도하면 로그인 페이지로 리다이렉트된다.

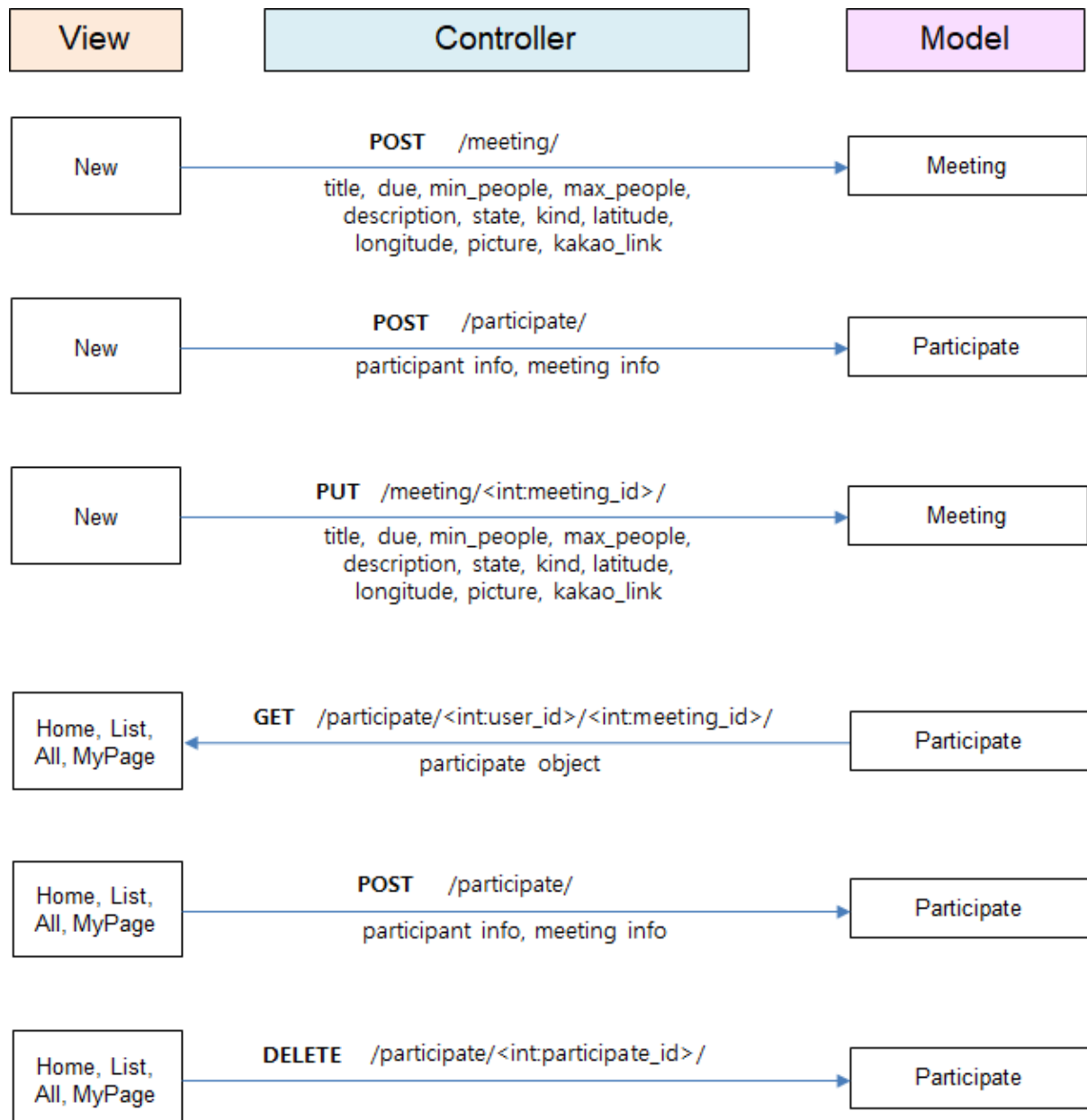
4) Controller

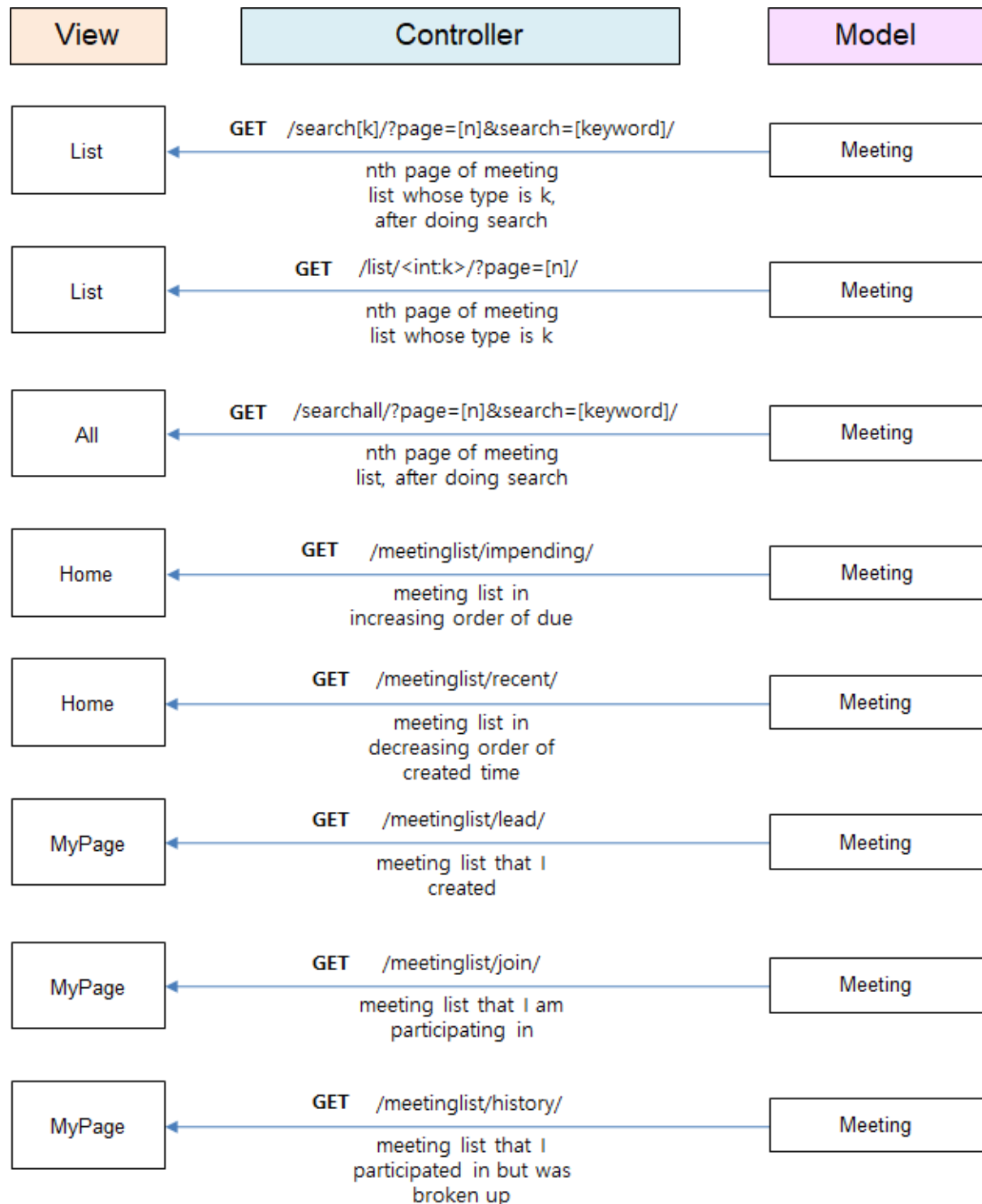
User Interface (UI)

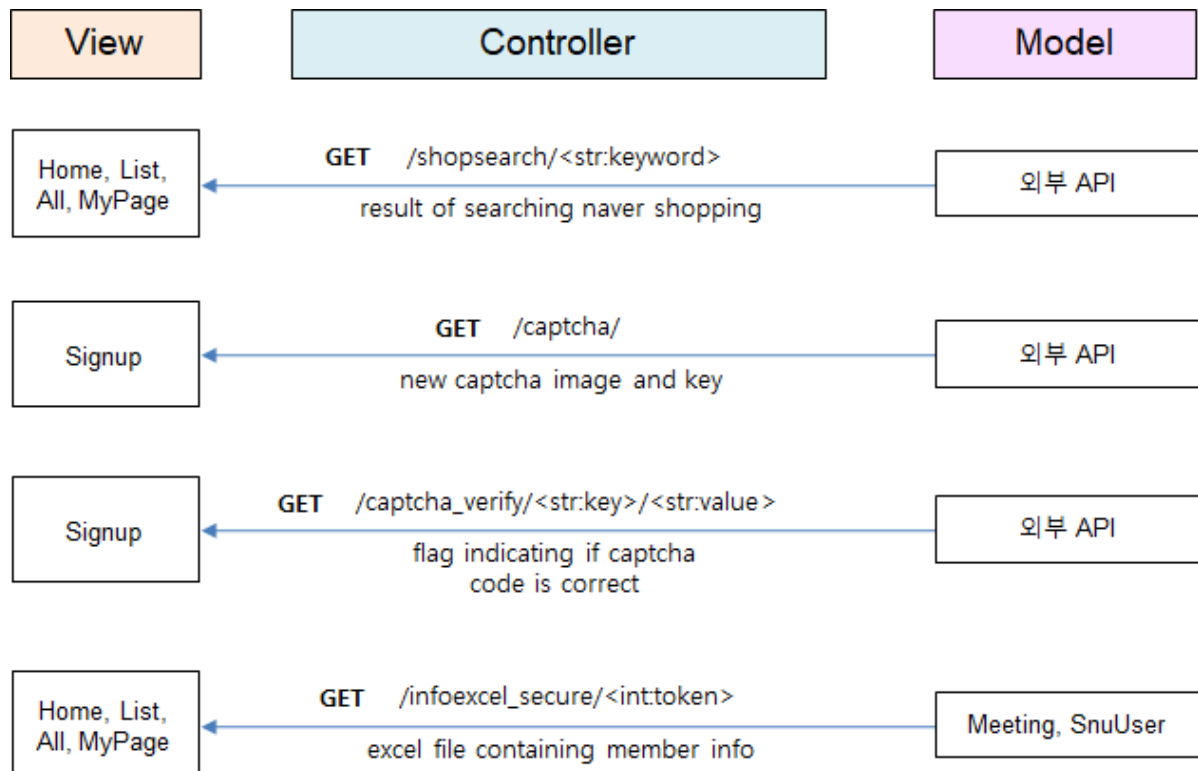
MVC 구조 중 컨트롤러를 디자인 하기 위한 도식은 아래와 같다. 오른쪽 화살표는 클라이언트 측에서 사용자가 HTTP Request 를 보냈을 때 프론트 엔드의 뷰가 컨트롤러를 통해 백 엔드의 모델을 생성, 수정, 혹은 삭제하는 것을 보여준다. 왼쪽 화살표는 백 엔드가 모델의 정보를 컨트롤러를 통해 프론트 엔드의 뷰로 되돌려주고 프론트 엔드의 뷰가 클라이언트 측의 사용자에게 HTTP Response 를 보내주는 것을 보여준다. 각 화살표의 위에 적혀 있는 주소는 프론트 엔드와 백 엔드 간에 JSON 데이터를 교환하기 위한 API 의 URL 에 해당한다.











Design Details

1) Front-end Design

Front-end Components

우리 웹 서비스는 Atomic React를 기반으로 프론트 엔드를 디자인했기 때문에 컴포넌트들을 아래와 같이 상위 계층부터 도식화하였다. 각 컴포넌트의 속성값 및 함수들은 해당 박스 안에 표시되어 있다.

<div>SignupPage</div> <div>Username: input Password: input Password Confirm: input name: input Captcha: Captcha 회원가입: button 돌아가기: button + signup_click</div>	<div>LoginPage</div> <div>Username: input Password: input 로그인: button 회원가입: button 홈페이지 소개: button + login_click</div> <div>AllPage</div> <div>Left_sidebar: Left_sidebar Right_sidebar: Right_sidebar AllList: AllList + change_page_num_click</div> <div>ListPage</div> <div>Left_sidebar: Left_sidebar Right_sidebar: Right_sidebar KindList: KindList keyword: input Search: button NaverShopping: NaverShopping + change_page_num_click</div> <div>TutorialPage</div> <div>To Home: button explanation: text</div>	<div>LoginAuthPage</div> <div>email: input email_code: input 인증번호 전송: button 확인: button phone_number: input phone_code: input 인증번호 전송: button 확인: button 인증완료: button 로그아웃: button 홈페이지 소개: button + prepare_send_email_click + send_email_click + confirm_email_click + prepare_send_phone_click + send_phone_click + confirm_phone_click + logout_click</div> <div>NewPage</div> <div>Left_sidebar: Left_sidebar Right_sidebar: Right_sidebar MeetingCreate: MeetingCreate</div>
--	---	--

MeetingInfo	ImpendingList	LeadList
accuse_content: text 완료: button 취소: button basic_info: text 정보보기: button 신고하기: button member_list: button 참여 멤버 정보 다운: button 오픈 채팅 링크: button picture: image Map: Map CloseButton: ButtonsInMeetingInfo BreakUpButton: ButtonsInMeetingInfo ModifyButton: ButtonsInMeetingInfo ReOpenButton: ButtonsInMeetingInfo ReCloseButton: ButtonsInMeetingInfo JoinButton: ButtonsInMeetingInfo WithdrawButton: ButtonsInMeetingInfo CommentList: CommentList + change_meeting_state_click + change_meeting_info_click + join_meeting_click + withdraw_meeting_click + prepare_load_leaderinfo_click + load_leaderinfo_click + prepare_load_memberinfo_click + load_memberinfo_click + accuse_click	label: text header: text 접기: button 펼치기: button MeetingEntry: MeetingEntry	label: text header: text 접기: button 펼치기: button MeetingEntry: MeetingEntry
	RecentList	JoinList
	label: text header: text 접기: button 펼치기: button MeetingEntry: MeetingEntry	label: text header: text 접기: button 펼치기: button MeetingEntry: MeetingEntry
	KindList	HistoryList
	label: text header: text 접기: button 펼치기: button MeetingEntry: MeetingEntry	label: text header: text 접기: button 펼치기: button MeetingEntry: MeetingEntry
	AllList	MeetingCreate
	label: text header: text 접기: button 펼치기: button MeetingEntry: MeetingEntry	meeting_info: text naverShopping: NaverShopping checkbox: radio 완료: button 수정: button + new_click + modify_click
Left_sidebar	Right_sidebar	Myinfo
To Home: button 홈페이지 소개: button 음식배달: button 택시합승: button 공동구매: button 스터디: button 운동: button 미팅: button 기타: button	keyword: input ToMyPage: ToMyPage Logout: Logout Search: button Make_new: Make_new Join_chatroom: Join_chatroom	My Information: text LeadList: LeadList JoinList: JoinList HistoryList: HistoryList
		ReportInfo
		description: text 처리: button 기각: button 되돌리기: button + penalty_click

Make_new	Map	Captcha
새 모임 만들기: button	map: daumMap API	image: image
		새로고침: button
Join_chatroom	ToMyPage	captcha code: input
title: text	내 정보: button	확인: button
kakao_link: button		+ prepare_load_captcha_click
		+ load_captcha_click
		+ confirm_captcha_click
MeetingEntry	lNaverShopping	ButtonsInMeetingInfo
info: text	title: text	ModifyButton: button
card: button	image: image	CloseButton: button
+ prepare_load_comments_click	price: text	BreakUpButton: button
+ load_comments_click	link: text	ReOpenButton: button
	닫기: button	ReCloseButton: button
	naver_symbol: image	JoinButton: button
Logout	query: input	WithdrawButton: button
로그아웃: button	Search: button	
+ logout_click	+ prepare_search_click	
	+ search_click	

Front-end Algorithms

1. SignupPage

- signup_click(username : string, password : string, name : string) : 사용자가 입력한 아이디, 패스워드, 이름을 백 엔드에 넘겨준다. 그래서 아이디가 기존에 존재하지 않는 것이라면 회원가입에 성공하고 인증 페이지로 이동한다.

2. LoginPage

- login_click(username : string, password : string) : 사용자가 입력한 정보(username, password)를 백 엔드에 넘겨준다. 로그인이 성공한다면 Home 페이지로 이동하고, 실패한다면 '존재하지 않는 ID나 비밀번호입니다.'라는 경고창을 띄운다.

3. LoginAuthPage

- logout_click() : Login 페이지로 이동한다.
- prepare_send_email_click(hash : string, email : string, email_code : string) : 이메일 인증 전송이 완료될 때까지 로딩 창을 띄우고, 이메일 인증 전송이 완료되면 로딩 창이 꺼진다.
- prepare_send_phone_click(hash : string, phone_number: string, phone_code : string) : 휴대폰 인증 전송이 완료될 때까지 로딩창을 띄우고, 휴대폰 인증 전송이 완료되면 로딩창이 꺼진다.

- send_email_click(hash : string, email : string) : MySNU 계정 메일인지 확인하고, MySNU 계정이라면 사용자가 입력한 해당 이메일로 인증번호를 보내고 해당 유저의 이메일 토큰 필드를 설정한다.
- send_phone_click(hash : string, phone_number : string) : 휴대폰 번호 형식을 확인하고, 형식이 일치한다면, 사용자가 입력한 해당 휴대폰 번호로 인증번호를 보내고 해당 유저의 휴대폰 토큰 필드를 설정한다.
- confirm_email_click(hash: string, email : string, email_code : string) : 사용자가 입력한 이메일 인증 코드가 백엔드에서 온 인증 토큰과 정확히 맞는지 확인한다.
- confirm_phone_click(hash: string, phone_number: string, phone_code : string) : 사용자가 입력한 휴대폰 인증 코드가 백엔드에서 온 인증 토큰과 정확히 맞는지 확인한다.

4. AllPage

- change_page_num_click(page_num : number) : All 페이지에서 page_num에 해당하는 페이지로의 이동을 시도한다.

5. ListPage

- change_page_num_click(page_num : number) : List 페이지에서 page_num에 해당하는 페이지로의 이동을 시도한다.

6. NaverShopping

- prepare_search_click() : 검색 결과가 올 때까지 로딩 창을 띄우고, 검색 결과가 온다면 결과 창으로 바뀐다.
- search_click(query : string) : 사용자가 입력한 키워드를 쇼핑 api가 제공한 url로 보낸다.

7. MeetingCreate

- new_click(hash : string, user_id : number, meeting_info : string) : 새로운 모임을 생성할 때 쓰인다. 모임 글을 쓴 유저가 누구인지, 모임의 제목, 마감기한, 모임 유형, 최소 인원, 최대 인원, 오픈 채팅 링크, 본문, 리더, 사진들을 백 엔드에 POST 하여 새로운 모임을 생성한다.
- modify_click(hash : string, meeting_info : string) : 기존 모임을 수정할 때 쓰인다. 모임의 제목, 사진, 본문 내용들을 백엔드에 PUT 하여 기존의 모임 글을 수정한다.

8. ReportInfo

- `penalty_click(hash : string, flag : bool, report_info : string, point : number)` : 운영자가 신고 목록들 중에서 하나를 처리했을 때, 쓰이는 함수이다. 신고를 처리 또는 기각할 경우와 신고를 되돌릴 경우로 나누며, 처리의 경우에는 신고 내용과 별점을 백엔드에 PUT해 별점을 부과한다. 기각할 경우에는 별점 0점을 부과한다. 그리고 되돌릴 경우에는 운영자가 처리한 별점 결과를 0점으로 바꾼다.

9. Captcha

- `prepare_load_captcha_click()` : 새로운 captcha 이미지를 받아오기 전까지 default 이미지를 보여주며, 이미지를 받아오면 default 이미지가 사라지고 받아온 이미지가 보인다.
- `load_captcha_click()` : 새로고침 버튼을 누르며 새로운 captcha 이미지를 요청한다.
- `confirm_captcha_click(key : string, value : string)` : captcha에서 받아온 key 값과 사용자가 쓴 결과를 비교한다. captcha api의 url에 키 값과 value 값을 보내면 보안문자를 맞게 입력했는지 확인할 수 있다.

10. MeetingInfo

- `change_meeting_state_click(hash : string, meeting_info: string, new_state : number)` : 모임의 상태를 변화시킨다. 모임의 상태에는 '모집 중', '마감', '추가 모집 중', '추가 모집 마감', '해산' 이 있으며, '모집 중'에서는 '마감', '마감'에서는 '추가 모집 중' 이런 식으로 다음 상태로 변화시킬 수 있으며, 모든 상태에서 '해산'으로 변화시킬 수 있다.
- `change_meeting_info_click(meeting_info : string)` : 모임 정보를 수정한다. New 페이지로 redirect 되어 모임을 수정할 수 있다.
- `join_meeting_click(hash : string, user_id : number, meeting_id : number)` : 해당 모임에 참가한다. '참가' button을 누른 사용자를 해당 모임에 참가시킨다.
- `withdraw_meeting_click()` : 해당 모임에서 탈퇴한다. '탈퇴' button을 누른 사용자를 해당 모임에서 탈퇴시킨다.
- `prepare_load_leaderinfo_click()` : 모임 주최자의 정보를 가져오기 전에 호출하여 정보를 가져오기 전까지 창이 뜨지 않도록 하여 전에 클릭했던 모임의 주최자 정보가 render되지 않게 한다.
- `load_leaderinfo_click(user_id : number)` : 모임 주최자의 정보(이름, 이메일, 전화번호, 별점)을 가져와서 Dropdown 형식으로 보여준다.
- `prepare_load_memberinfo_click()` : 모임 참여자의 정보를 가져오기 전에 호출하여 정보를 가져오기 전까지 창이 뜨지 않도록 하여 전에 클릭했던 모임의 참여자 정보가 render되지 않게 한다.
- `load_memberinfo_click(numbers : number array)` : 모임 참여자의 정보(이름, 이메일, 전화번호, 별점)을 가져와서 Dropdown 형식으로 보여준다. 참여자 정보를 보기 위한 함수들을 호출하는

button은 오직 모임 주최자만이 누를 수 있다.

- `accuse_click(hash : string, accuse_reason : string, member_id : number)` : 해당 유저를 신고한다. 신고 내용을 작성하고 완료/취소를 누를 수 있는 신고 창이 나타난다. 신고는 운영자가 `ReportAdminPage`에서 읽어보고 처리할 수 있다.

11. MeetingEntry

- `prepare_load_comments_click()` : 댓글 정보를 가져오기 전에 호출하여 댓글을 가져오기 전까지 빈 창이 보이게 하여 전 댓글 정보가 render 되는 것을 막는다.
- `load_comments_click(meeting_id : number)` : 해당 모임의 댓글들을 가져와서 comment 란에 보여준다.

2) Back-end Design

Django RESTful API

백 엔드에서는 앞서 MVC 모델에서 논의한 것을 기반으로 모델들을 디자인하였으며, 그러한 모델들을 이용하여 프론트 엔드와 통신하기 위해 설계한 Django RESTful API는 다음과 같다.

Model	API	GET	POST	PUT	DELETE
SnuUser	/log_in/	get logged user info	X	X	X
	/sign_up/	X	create a new account	X	X
	/send_email/ <str:email>/	send authentication code to email	X	X	X
	/send_phone/ <str:phone>/	send authentication code to phone	X	X	X
	/email_auth/ <str:email>/ <str:email_code>	confirm email authentication code	X	X	X
	/email_auth/ <str:phone>/ <str:phone_code>	confirm phone authentication code	X	X	X

	/user/ <int:user_id>	get user info	X	X	X
Report	/report/	get report list	accuse a user	X	X
	/report/ <int:report_id>	X	X	handle a report	X
Comment	/comment/	X	add a comment	X	X
	/comment/ <int:comment_id>	X	X	edit a comment	delete a comment
	/comment/ meeting/ <int:meeting_id>	get a comment list on a meeting	X	X	X
Meeting	/meeting/	X	create a new meeting	X	X
	/meeting/ <int:meeting_id>/	X	X	modify a meeting post	X
	/search[k]/ ?page=[n] &search=[keyword]/	get nth page of meeting list whose type is k, after doing search	X	X	X
	/list/<int:k>/ ?page=[n]/	get nth page of meeting list whose type is k	X	X	X
	/searchall/ ?page=[n]& search=[keyword]/	get nth page of meeting list, after doing search	X	X	X
	/meetinglist/ impending/	get meeting list in increasing order of due	X	X	X
	/meetinglist/ recent/	get meeting list in decreasing order of created time	X	X	X
	/meetinglist/ lead/	get meeting list that I created	X	X	X
	/meetinglist/ join/	get meeting list that I am participating in	X	X	X
	/meetinglist/ history/	get meeting list that I participated in but was broken up	X	X	X

Participate	/participate/	X	join a meeting	X	X
	/participate/ <int:participate_id>	X	X	X	withdraw from a meeting
	/participate/ <int:user_id>/ <int:meeting_id>	get a participate object with user_id and meeting_id	X	X	X
기타	/shopsearch/ <str:keyword>	search naver shopping (External API)	X	X	X
	/captcha/	get captcha image and key (External API)	X	X	X
	/capthca_verify/ <str:key>/ <str:value>	confirm captcha code (External API)	X	X	X
	/infoexcel_secure/ <int:token>	get a excel file containing member info of a meeting with the specified token	X	X	X