

3on3

김요한 박진우 진상훈
김은지 문지은 이예진

과감하게 링기자 땠고



목차

주제	기능
주제선정이유 4	기능시연 21
요구사항정의 6	기능소개 22
	트러블 슈팅 31

개발	향후 업데이트
개발의도 및 기대효과 8	향후 업데이트 38
개발일정 9	한줄 소감 39
개발도구 11	
어플리케이션 모델링 12	
아키텍처 소개 14	
화면정의서 15	
팀역할 및 담당 19	



GitHub 바로가기

주제

go

대학생 80.7% 나는 '코로나 학번'

신승희 기자

대학생 대부분(91.0%)이 대학 생활 중 코로나19의 영향을 받았다고 답했으며, 이들 5명 중 4명(84.2%)은 그 영향이 대체로 부정적이었다고 말했다. 구체적으로는 '동아리, 축제, OT 및 MT 등 대면 캠퍼스 활동을 누리지 못한 것(77.2%, 복수응답)'을 가장 많이 짚었고, 이 외에 ▲대학 동기, 선·후배 친목 도모 및 소통 부족(68.7%) ▲지속된 비대면 수업으로 수업의 질 하락(60.8%) ▲학교 시설물 및 기자재 사용의 어려움(34.3%) ▲교환학생, 해외 봉사 등

대학생 커뮤니티에서 싹트는 사랑



익명

11/12 11:21

소개팅 주선

크리스마스가 얼마 안남았어요~
에타에 외로워하는 분들이 많아서 한번 모아봅니다..!



현재 대학중인 분들 모두 참여 가능합니다!
쪽지로 양식대로 보내주세요!

1. 나이, 성별
2. 단과대 (같은 단대 피하려구)
3. 자신의 매력 2가지 (외모 제외)
4. 상대방에게 원하는 매력 2가지 (외모 제외)
5. 요구사항 (술담배 하는 사람은 싫다~ 나이가 같았으면 좋겠다 등등) (안적어도 상관x)

★쪽지로 보내주시면 잘 맞는 분들끼리 오픈채팅방
만들어드릴게요~~
좀 모집되면, 댓글로 진행상황 알려드릴게요!

☆매칭 후 저는 아무런 책임도 지지 않습니다..!

예전에 올라왔던 모집글 참고했어요!
많이 참여해주세요~~

9 12 2

■근 대학생 커뮤니티 사이트'에
브리타임'에서 미와 같은 비대면
만남을 주선하는 게시물이 등장하고 있다.



© 에브리타임

트래일러 브리타임

코로나19 팬데믹 2년, 달라진 대학 연애풍속도

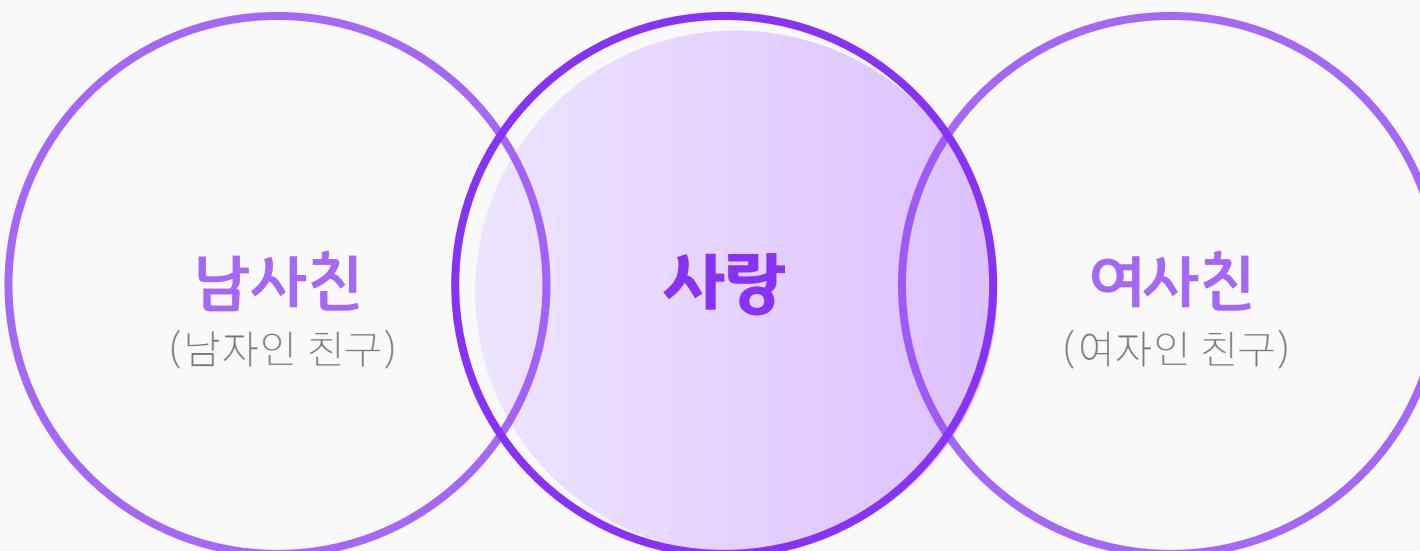
'에브리타임', '캠퍼스픽', '소개팅앱' 등 다양한 비대면 만남 등장

21.12.07 13:25 | 최종 업데이트

반면 같은 학교에 재학 중인 박현수(24·가명)씨는 "서로 간단히 이야기와 사진을 교환하고 친해질 수 있는 발판을 온라인을 통해 마련하고 만난다면 실제로 만났을 때 더 깊고 많은 주제로 이야기를 많이 할 수 있을 것 같다"며 긍정적인 입장을 밝혔다. 또한 "나도 이러한 게시물을 통해 이야기를 나눠보고 비슷하다고 느낀다면 한 번 정도는 해보고 싶다"고 말했다.

해당 대학교에 재학 중인 또 다른 학생 김지수(24·가명)씨는 "에브리타임은 같은 대학 학생이라는 보장이 있는 만큼 이곳에서 비대면 소개팅을 진행해 만나 연애하는 것에 대해 나쁘게 생각하지 않는다"라고 답했다.

주제 선정 이유



새로운 만남 추구 & 사랑

대학생을 주타겟으로 선정, 관심사 탐색

대면 소통 기회가 사라진 요즘 대학생들

코로나 전후로 바뀐 대학생활 체감

코로나 세대 대학생들을
타겟으로 소개팅 어플 개발!

요구사항 정의

평가 및 리뷰

비싸다

★★★★★

대학생을 타깃으로 하면 돈이 없다는 것을 알텐데, 1회 소개팅에 14,300원을 낸다는 것이 너무 비싼 것 같다. 만원이었던 시기가 있었던 것 같은데, 사실상 43%가 오른 셈이다. 만원이라면 많이 했을 것 같다. 특히 처음으로 소개팅/미팅을 시도해보는 사람한테는 '보증되지 않은 서비스와 상대방' 그리고 14,300원을 비교하게 된다. 신뢰가 확보되지 않은 첫 가입자에게는 마음을 접게 만드는 요소라고 생각된다. 예전에 유튜브 프리미엄 3개월 무료를 진행했던 것처럼, 첫 가입자에게는 반값 할인 등을 통해서 마음을 사야할 것이다.

나이 수정

★★★★★

나이를 잘못 입력했는데 수정도 안되고 탈퇴도 안되네요;;

평가 및 리뷰

새로운 어플이라 신선향용

★★★★★

다 좋은데 팀 만들고 나서 팀원들 수정 가능하면 좋을것같아용.. 수정이 안되어 불편해요 ㅜㅜ

개발자 답변

1월 23일

안녕하세요!! 이 모이는 곳입니다. 에 관심을 가지고 소중한 의견주셔서 감사합니다 😊 저희 앱을 좋게봐주시고 사용해주셔서 감사드립니다...!! 앞으로도 대학생들을 위한 컨텐츠를 제공할 수 있도록 열심히 노력하겠습니다. 즐거운 하루 되시길 바랍니다 🙏

결제하고선 매칭 한번도 안됨

★★★★★

10000원 넣고 수십번 중 최종 매칭 한번도 안됨. 거절을 3번해야 하트를 하나 받아서 여자의 경우 거의 거절함. 제공되는 프로필 정보도 너무 적고 사진 한장만 보고 선택해야하는 단점. 남자의 경우 자기가 잘생긴 정도가 아니고 진짜 잘생긴거 아니면 하시 마세요. 돈 아깝다..

아니 대체

1년 전

★★★★★

어떻게 탈퇴하라는거야? 원 놈의 어플이 고객센터도 없어



대학생이 타겟이기 때문에 안정성과 신뢰성 보장을 위해서 대학생 신분이 더 정확하게 인증되어야 한다.



너무 많은 정보 입력창은 사용자로 하여금 부정적인 인식을 갖게 하기 때문에 최소한의 정보만 입력 받아야 한다.



매칭이 실제 만남까지 성사되도록 유도하고, 사용자 유치와 접속시간 연장을 위해 어플리케이션 내부에서 매칭된 사용자들끼리 소통이 가능해야 한다.

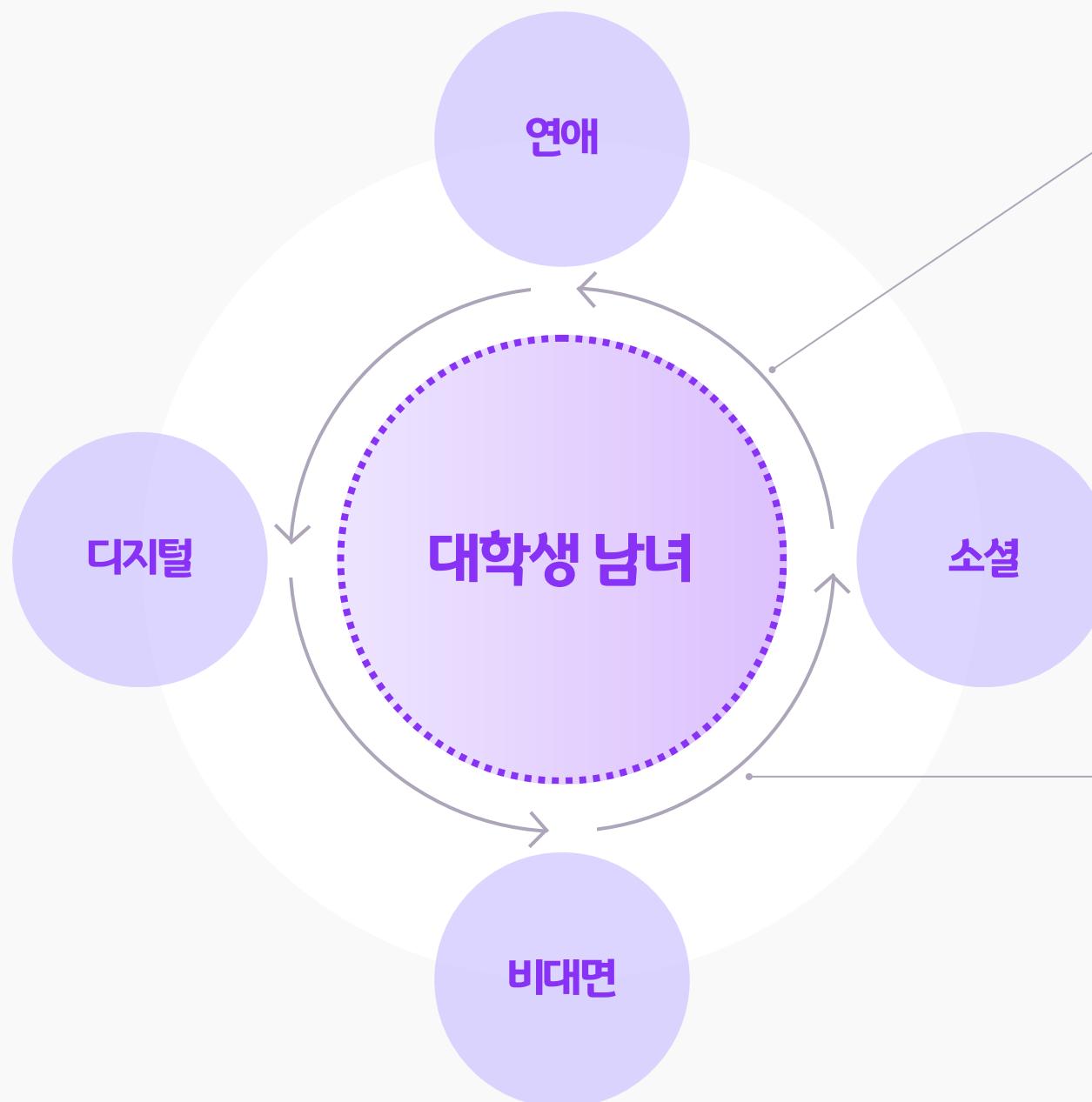


사용자가 어플 구조를 쉽게 이해할 수 있어야하고, 그룹을 생성하고 미팅을 하기까지 과정이 단순하게 진행되어야 한다.

개발



개발의도 및 기대효과



개발 배경

현재 대학생들은 온라인 플랫폼을 통해 관계를 맺는 소셜 미디어나 네트워킹 어플에 익숙하다. 또한 최근 몇 년간의 팬데믹으로 인해 비대면 수업과 활동이 늘어남에 따라 이 문제는 더 심화되었다.



기대 효과

소개팅 어플 '과팅'은 해외의 틴더 등과 같은 소개팅 어플리케이션을 벤치마킹하여 한국의 대학생들을 주요 타겟층으로 소통과 만남을 증가한다. 이 어플은 학교나 학과 기반의 네트워크를 형성할 수 있고, 학생들이 좀 더 신뢰할 수 있는 인맥을 구축하는데 도움을 준다.

개발 일정 | 7월

Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17 주제 선정 & 기능회의	18	19 ERD	20
21	22	23	24 마크업 + 회의	25	26	27
			디자인			
28	29	30	31			
		개발 로그인 & 회원가입				

개발 일정 | 8월

Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
				1 기능 개발	2	3
4	5	6	7	8	9	10
11	12	13	14 신기능 회의	15 신기능 기능 개발 & 추가ERD 작업	16	17
18	19	20	21	22 배포 + 발표 준비 + 회의	23	24
	버그 픽스					
25	26	27	28	29	30	31

개발 도구

FRONTEND



BACKEND



DB & 배포

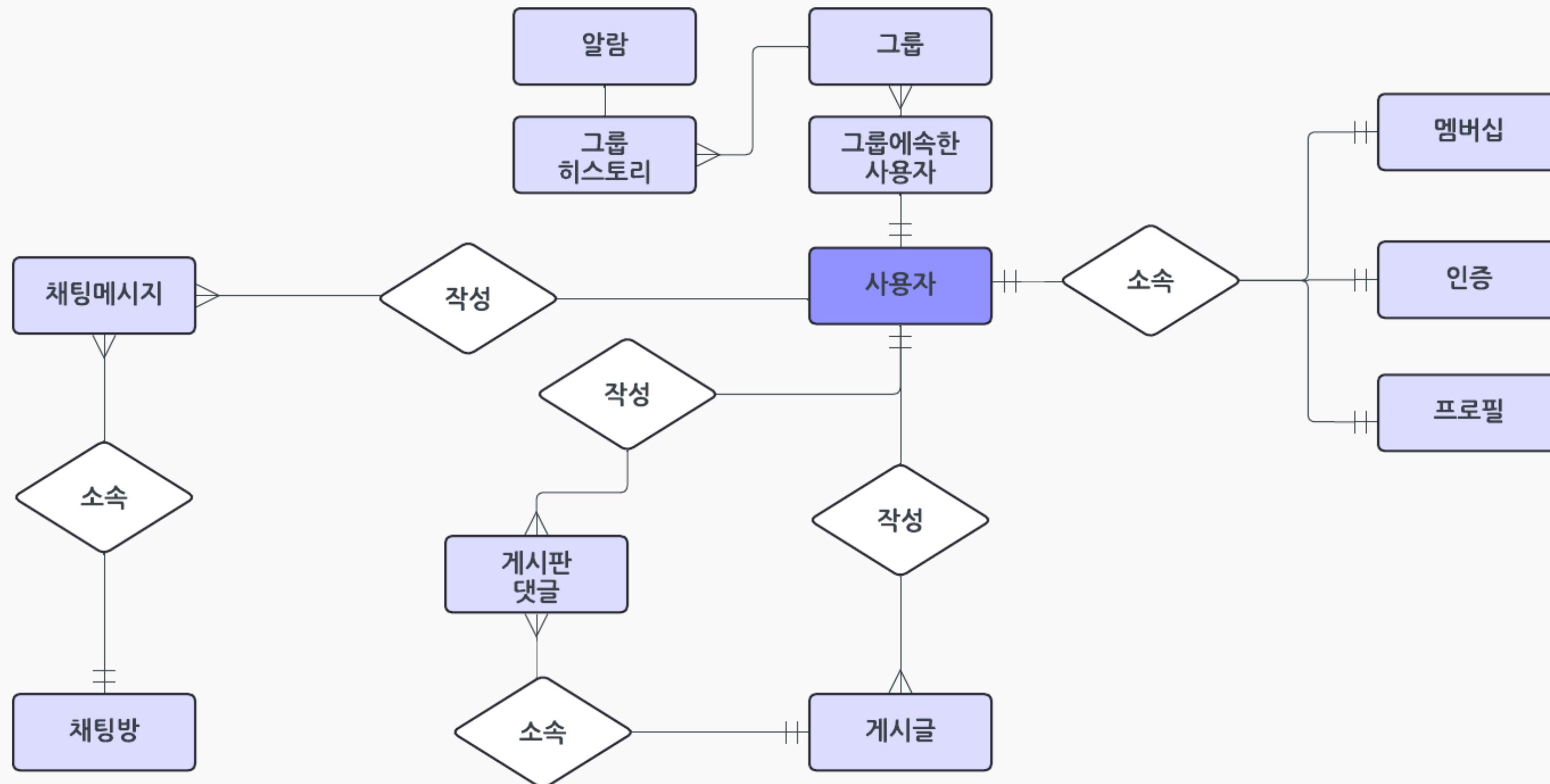


개발 환경



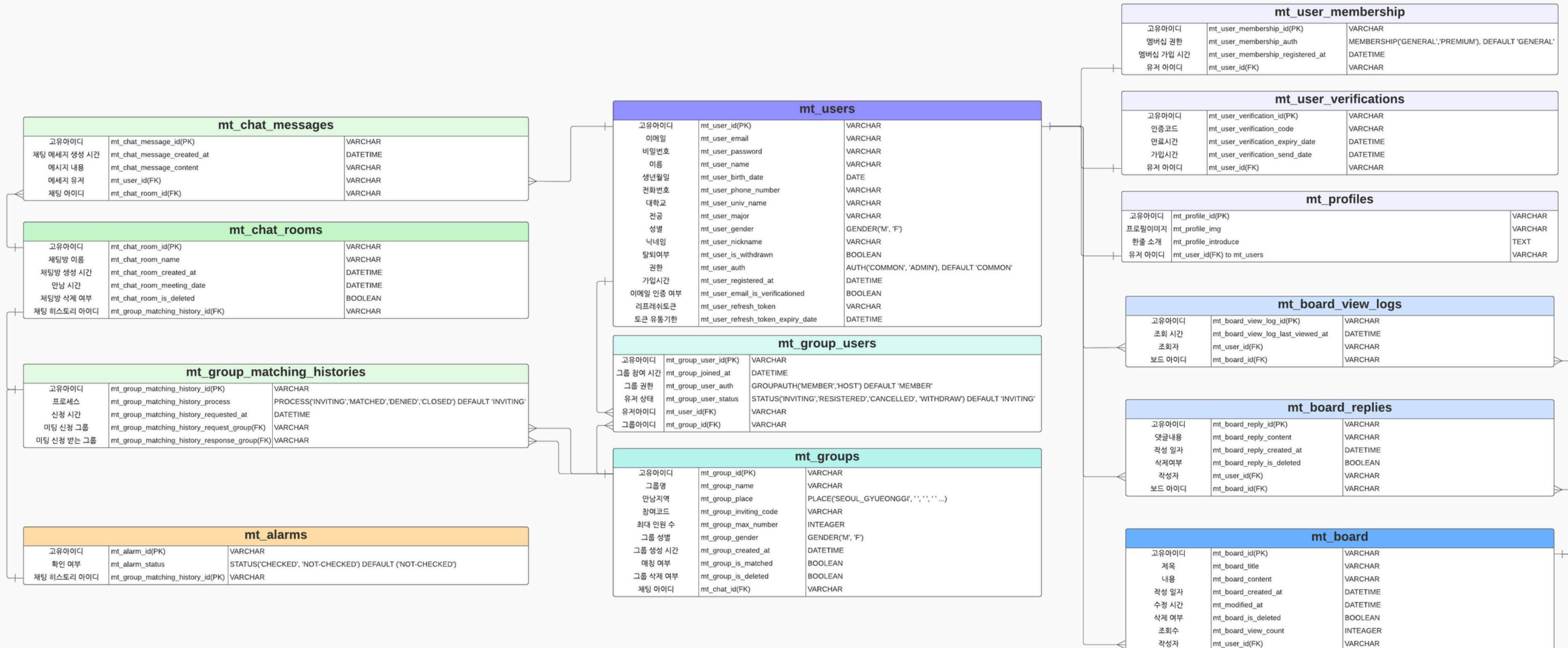
어플리케이션 모델링

| 개념적 모델링

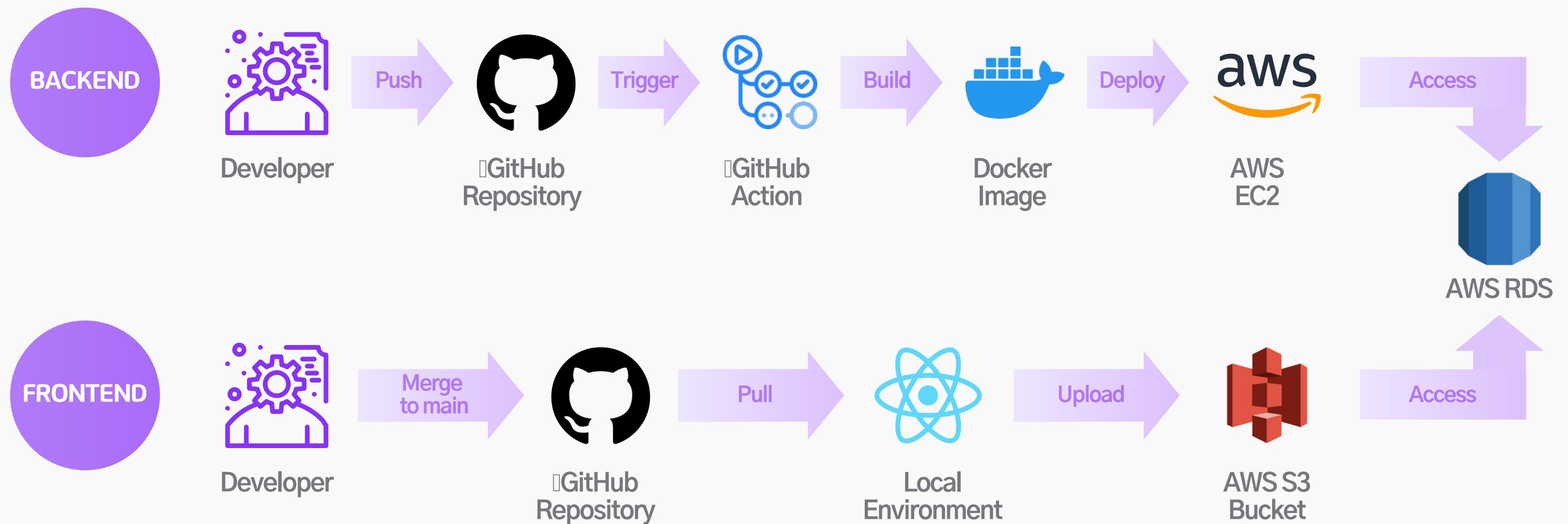


어플리케이션 모델링

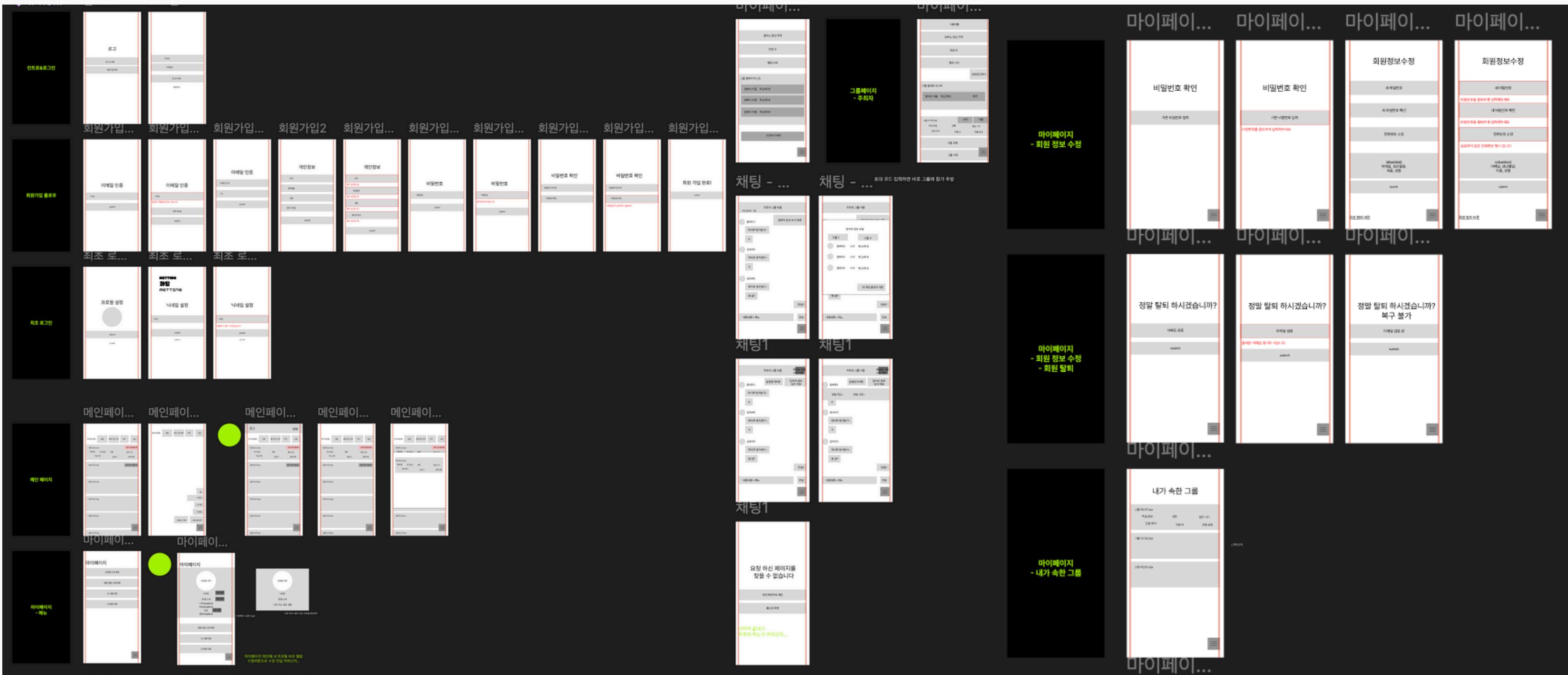
| 물리적 모델링



아키텍처 소개



화면정의서 | 피그마



화면정의서 | 노션

회의

회의 캘린더 유형별 리스트 내 회의 전체 +

↑ ↓ ⚡ 🔎

시나리오

1. 1차 회의(@2024년 7월 14일) 팀 주간 회의
2. 2차 회의(@2024년 7월 17일) 짧은 회의
3. 3차 회의 (@2024년 08월 09일)
4. 4차 회의(@2024년 08월 13일)
5. 5차 회의 (@2024년 08월 14일)
6. PPT 내용 정리(@2024년 08월 25일 ~ 2024년 08월 27일)
7. PPT 대본(@2024년 08월 27일)

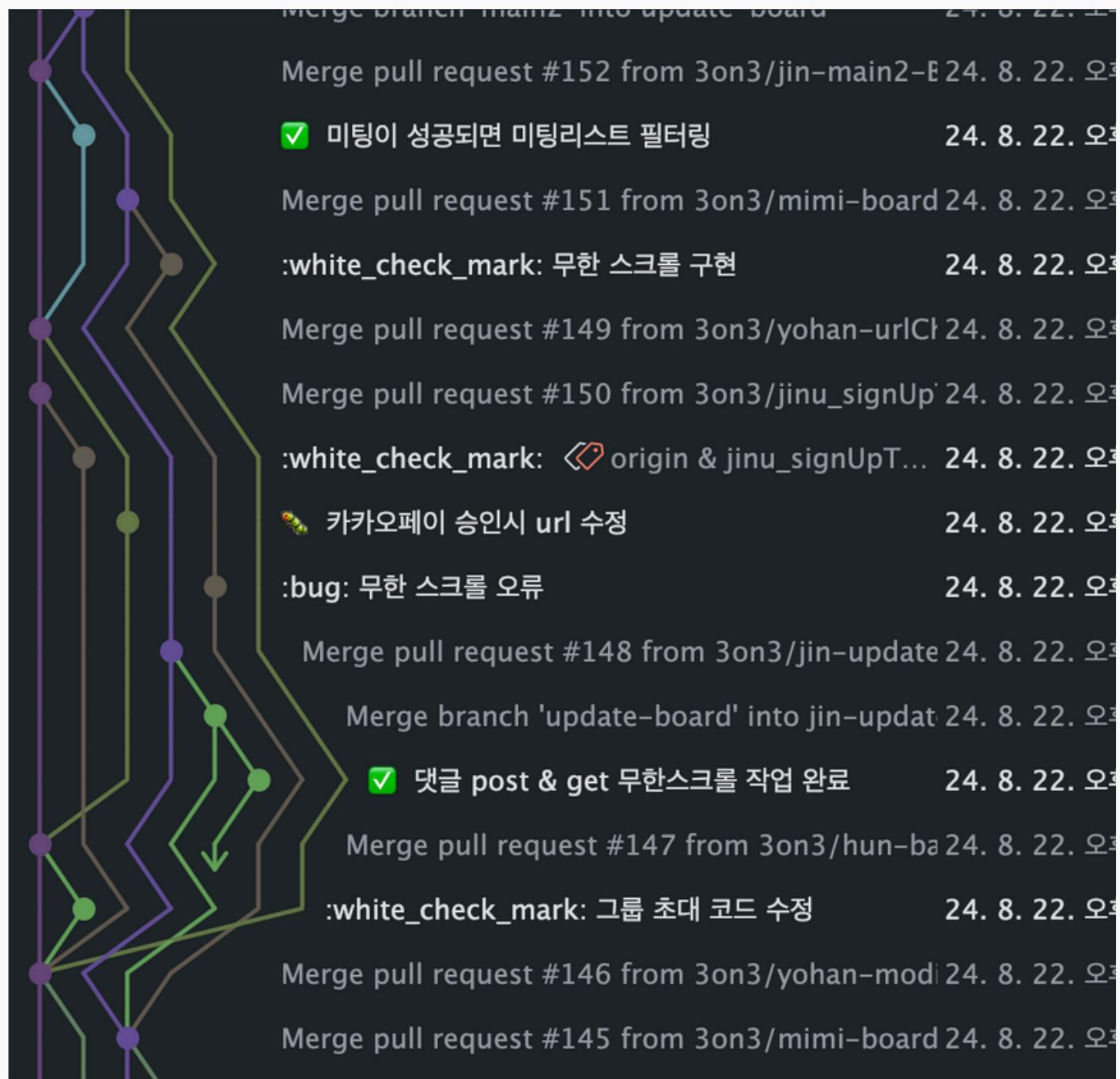
화면정의서 | 노션

01-1	로그인 / 회원관리	회원가입	<ul style="list-style-type: none"> - 이메일 - 1 <ul style="list-style-type: none"> - 학교 - 1 - 전공 - 2 - 이름 - 3 - 생년월일 - 3 - 성별 - 3 - 전화번호 - 3 - 비밀번호 - 4 	<ul style="list-style-type: none"> - 이메일: 최초 학교 이메일 인증 후 인증 받은 이메일로 아이디 입력 대체 - 닉네임: 이름으로 표시 안하고 닉네임으로 표시함. - 비밀번호: 소문자, 대문자, 숫자, 특수문자 합해서 8~16자 (!, @, #, \$, %, ^, &, *) - 처음 로그인 시 프로필, 닉네임 설정 유도 페이지로 이동(스킵 가능) 	mt_users / 테이블명	<ul style="list-style-type: none"> - 고유 아이디: mt_user_id(PK) - 이메일: mt_user_email - 비밀번호: my_user_password - 이름: mt_user_name - 생년월일: mt_user_birth - 학교: mt_user_univ - 전공: mt_user_major - 성별: mt_user_gender - 닉네임: mt_user_nickname - 탈퇴여부: mt_user_is_withdrawn - 권한: mt_user_auth (common/premium/admin) - 가입시간: mt_user_registered_at - 전화번호: mt_user_phone_number 	X	<ul style="list-style-type: none"> • [로그인] UnivCert API를 활용한 이메일 인증 https://univcert.com/ ☺김요한 □ 배포시 univ_check → true로 □ User에서 나중에 univ → univName으로 바꾼거 적용하기 <pre>public void processSignUp(String email, String univName) { 1개 사용 위치 ↪yocong 1* try { // Step 1: API 호출을 통해 인증 메일 발송 Map<String, Object> response = UnivCert.certify(univCertApiKey, email, univName, univ_check: false); // Step 2: 인증 메일 발송 성공 시 임시 회원가입 }</pre>
01-2	로그인 / 회원관리	로그인	<ul style="list-style-type: none"> - 이메일 - 비밀번호 	자동로그인	mt_users / 테이블명	X		
01-3	로그인 / 회원관리	아이디 찾기	X		mt_users / 테이블명	X		
01-4	로그인 / 회원관리	비밀번호 찾기	<ul style="list-style-type: none"> - 이메일 - 새 비밀번호 - 새 비밀번호 확인 	<ul style="list-style-type: none"> - 이메일 인증 후 비밀번호 수정 - 수정 후 로그인화면으로 리다이렉트 	mt_users / 테이블명	X	<ul style="list-style-type: none"> • [마이페이지] 개인 정보 조회, 개인정보 수정 ☺김은지 	
01-5	로그인 / 회원관리	로그아웃	<ul style="list-style-type: none"> - 로그아웃 시 메인 페이지로 이동 	<ul style="list-style-type: none"> - 로그아웃 후 홈화면으로 리다이렉트 	mt_users / 테이블명	X	<ul style="list-style-type: none"> • [그룹] 그룹 생성, 그룹별 초대코드생성 ☺진상훈 	
02-1	마이페이지	프로필	<ul style="list-style-type: none"> - 프로필 이미지(이미지 파일) - 한 줄 소개 - 닉네임 - 이메일 - 생년월일 - 이름 - 성별 	<ul style="list-style-type: none"> - 수정 불가 정보 disabled 처리 	mt_profiles		<ul style="list-style-type: none"> • [그룹] 그룹 목록 조회, 필터링 ☺이예진 	
02-3	마이페이지	회원 탈퇴 (눈에 안띄게)	<ul style="list-style-type: none"> - 비밀번호 검증 	<ul style="list-style-type: none"> - 탈퇴여부 true로 전환 - 회원 탈퇴 후 홈 화면으로 리다이렉트 	mt_users	X	<ul style="list-style-type: none"> • [그룹] 그룹 매칭 신청 → [채팅] 채팅방 생성, Websocket.io를 활용한 채팅 서버 오픈 ☺문지은 	
02-4	마이페이지	내가 속한 그룹 (최대 3개)	<ul style="list-style-type: none"> - 그룹 페이지 노출 			X	<p>💡 채팅방 생성 로직</p> <ol style="list-style-type: none"> (신청자) 그룹에서 채팅 신청 버튼 클릭 <ol style="list-style-type: none"> 매칭 신청 불가 조건 <ol style="list-style-type: none"> 본인이 속한 그룹일 때 그룹 인원 수를 다 채우지 못 한 경우 이미 매칭 중인 경우 (+매칭 히스토리 denied 상태일 경우) 인원 수 다를 경우 희망지역 다를 경우 매칭 히스토리 생성 (주최자) 신청자 리스트에서 신청 수락 또는 거절 주최자 - 수락 <ol style="list-style-type: none"> mt_group_matching_history_process MATCHED 프로세싱 업데이트 그룹 간 채팅 방 생성 주최자 - 거절 <ol style="list-style-type: none"> mt_group_matching_history_process DENIED로 프로세싱 업데이트 	
03-1	멤버쉽	권한 과금	<ul style="list-style-type: none"> - COMMON 그룹 3개 - PREMIUM 그룹 5개 - 9900원 → 4900원(월정액) 	?????		X		
04-1	메인페이지(index)	미팅 리스트 그룹 생성/참여 미팅 필터	<ul style="list-style-type: none"> - 미팅 리스트 노출 - 그룹 생성 버튼 - 그룹 참여 버튼 - 위치별 필터링 버튼 	<ul style="list-style-type: none"> - 미팅 리스트 - 미팅 생성 버튼 리스트 생성, 참여코드 생성 - 미팅 참여 버튼 참여코드 입력 후 참여 		X		
04-2	메인페이지(미팅리스트)	그룹 매칭	<ul style="list-style-type: none"> - 그룹 생성 완료시 미팅 생성 or 미팅 신청 			-		
04-3	메인페이지(미팅리스트)	미팅 생성	<ul style="list-style-type: none"> - 미팅 생성 시 만남을 원하는 지역(8도) 선택 			X		
04-4	메인페이지(미팅리스트)	미팅 신청	<ul style="list-style-type: none"> - 생성된 그룹이 있으면 미팅 신청 가능 - 검색 or 필터를 통해 원하는 지역의 미팅 검색 가능 - 그룹 인원이 같아야 신청 가능 			X	<ul style="list-style-type: none"> • [채팅] Websocket.io 라이브러리를 활용한 메세지 전송 ☺박진우 	



노션 바로가기

화면정의서 | GitHub



Main branch

안정적이고 배포 가능한 상태의 코드가 담겨있는 운영 브랜치

Main2 branch

개발 중인 코드를 포함한 개발 브랜치

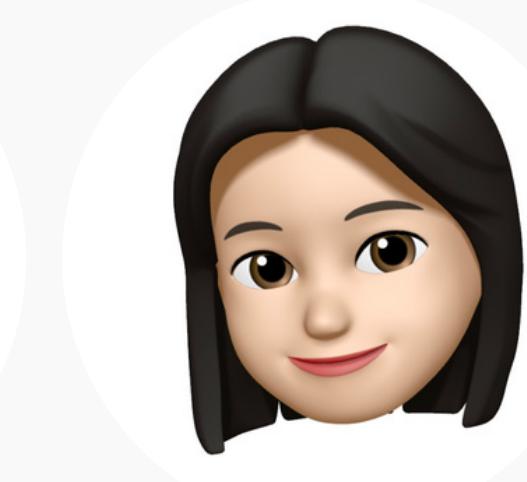
Update branch

새로운 기능을 개발할 때 사용하는 브랜치 각 기능은 별도의 브랜치로 분리되어 개발, 개발되면 main2 브랜치로 병합

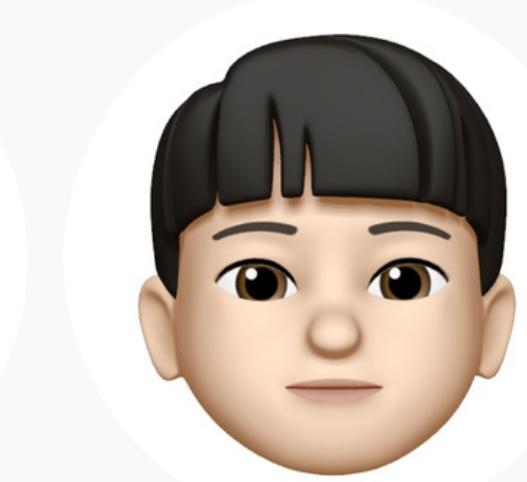
팀 역할 및 담당



팀장 | 진상훈



팀원 | 김은지



팀원 | 김요한



팀원 | 문지은



팀원 | 박진우



팀원 | 이예진

미팅 그룹 생성 / 삭제
초대코드 생성
참여자
(수락 / 거절 / 추방)
회원 탈퇴
이메일 인증
배포 Docker

마이페이지 회원 정보
(조회 / 수정)
프로필 이미지
(조회 / 수정)

대학 이메일 인증
회원가입
로그인 / 자동로그인
멤버십
(조회 / 결제)

그룹 매칭
(신청 / 수락 / 거절 / 닫힘)
채팅 방 생성
게시판
(조회 / 생성 / 수정 / 삭제)

실시간 채팅
채팅 방
(조회 / 삭제)
채팅 참여 유저 조회
매칭 알림 조회
실시간 매칭 알림

미팅 그룹 조회
다중 필터링
게시판 덧글
(생성 / 삭제)
무한스크롤

기능



9:21

LTE 87

학교 이메일 인증

건국대학교

mjsu10@konkuk.ac.

확인

보안 안 됨 — gwating.com

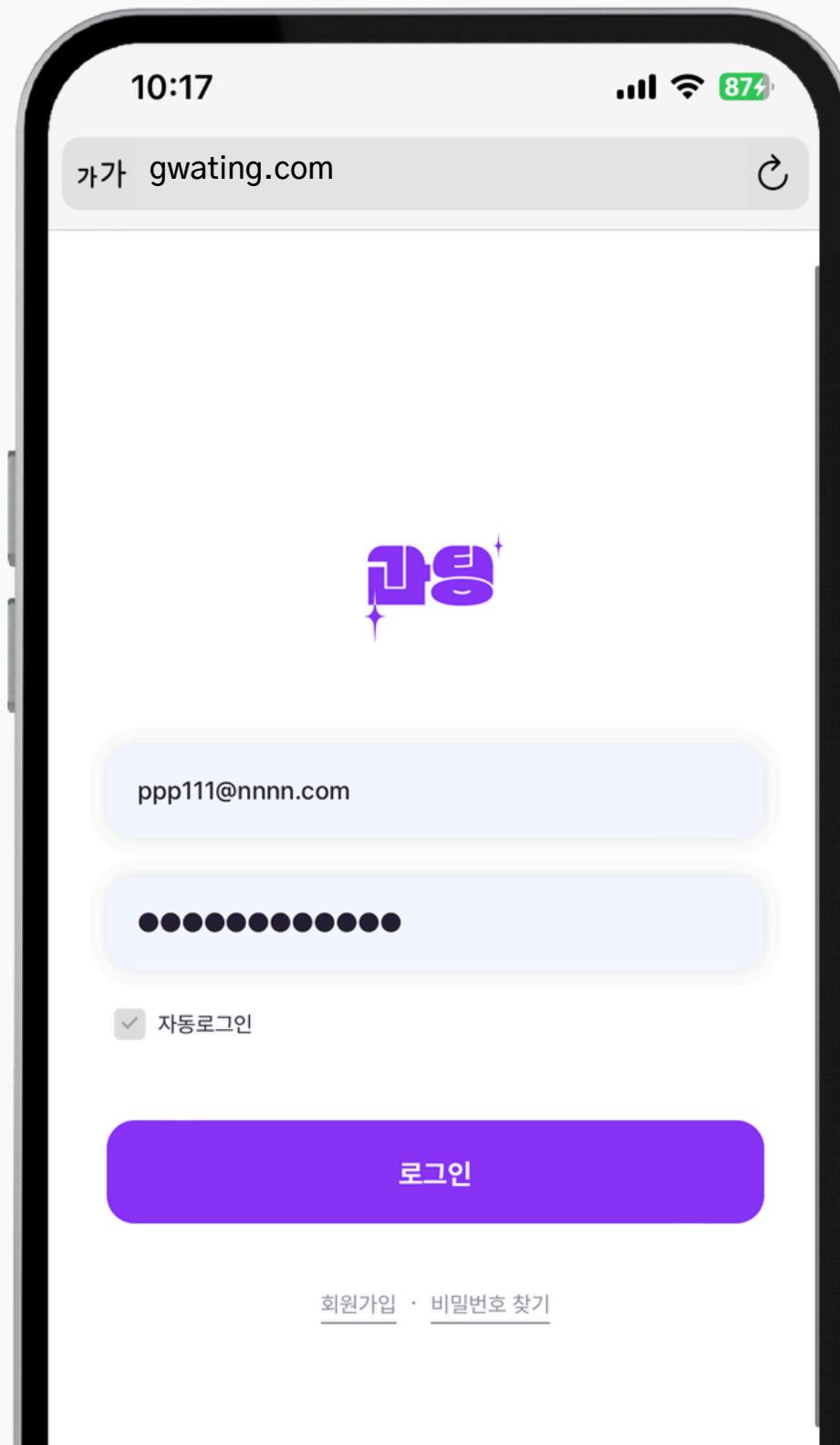
완료

1	2	3	4	5	6	7	8	9	0
-	.	;	()	₩	&	@	"		
#+=	,	?	!	'					✖
ABC	😊	space		return					



과팅 바로가기

기능 소개 | 회원가입 및 로그인

**이메일 인증**

대학메일인증 API를 활용하여 이메일 인증코드 검증
이메일 중복 확인 가능

회원 정보

전공, 이름, 생년월일, 성별, 비밀번호 입력
생년월일은 20대만 가능하게 필터링

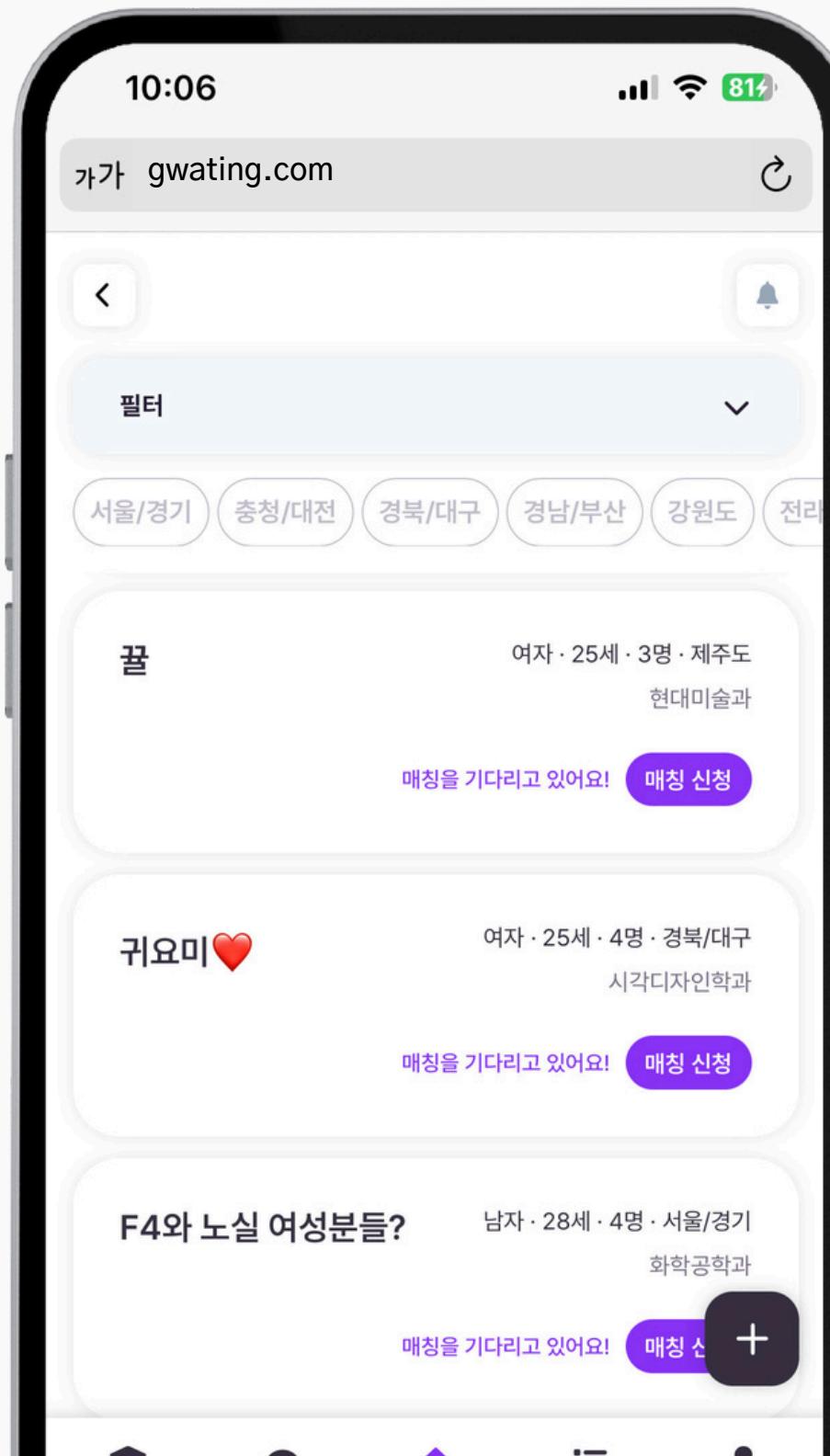
비밀번호 찾기

회원가입한 이메일 검증
새로운 비밀번호 설정하여 비밀번호 변경

로그인

자동 로그인 여부 체크 시 리프레쉬 토큰 발급
아이디, 비밀번호 중복 확인 및 존재여부 확인
최초 로그인시 프로필, 닉네임 설정 가능
설정하지 않을 시 기본 프로필, 랜덤 닉네임 부여
닉네임 중복 확인

기능 소개 | 메인페이지



과팅 목록 조회

다중 필터링

매칭 신청

무한스크롤

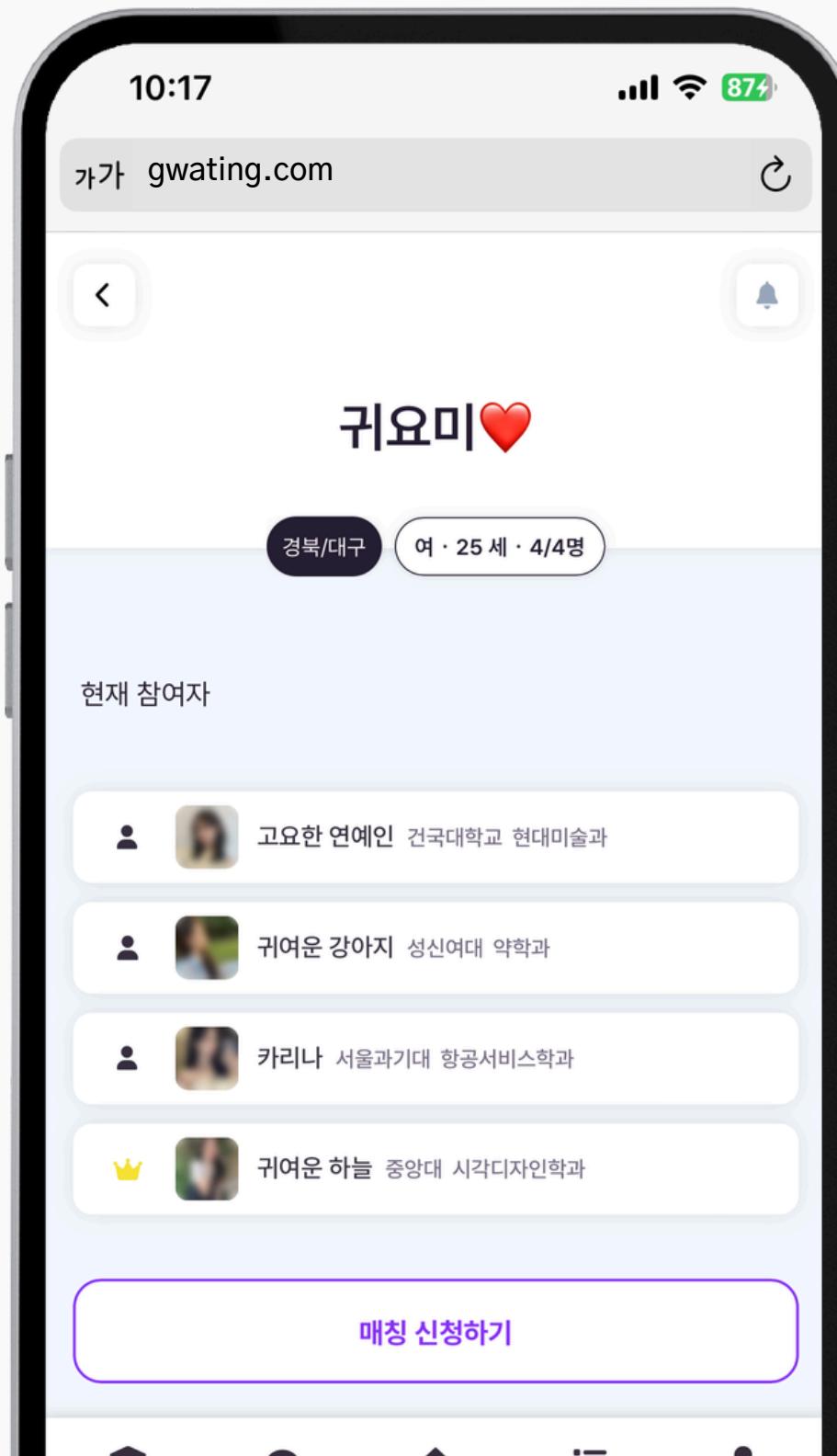
미팅을 원하는 사용자 그룹 목록 조회 가능
미팅이 성사된 그룹과 내가 속한 그룹은 제거
그룹 클릭 시 상세페이지로 이동

성별 & 참여 인원 (3:3 / 4:4 / 5:5) & 원하는 지역 선택 시
해당 조건에 따라 필터링 된 그룹 목록만 열람 가능

매칭 신청 버튼 클릭 시 빠르게 주최자 그룹에
매칭 신청 가능

최신 등록 순으로 무한스크롤 가능

기능 소개 | 그룹생성 & 그룹페이지



그룹 페이지

그룹 생성

초대코드 발송

그룹초대

그룹의 상세정보 확인 기능

참여자별 상태 (초대중 / 참여수락 / 참여거절) 확인 가능

설정 메뉴를 통해 초대코드 복사, 그룹 삭제, 참여자 추방 기능

미팅을 주최하고 싶은 사용자가 미팅 그룹을 생성

미팅 생성시 옵션으로 그룹 이름, 지역, 미팅 인원

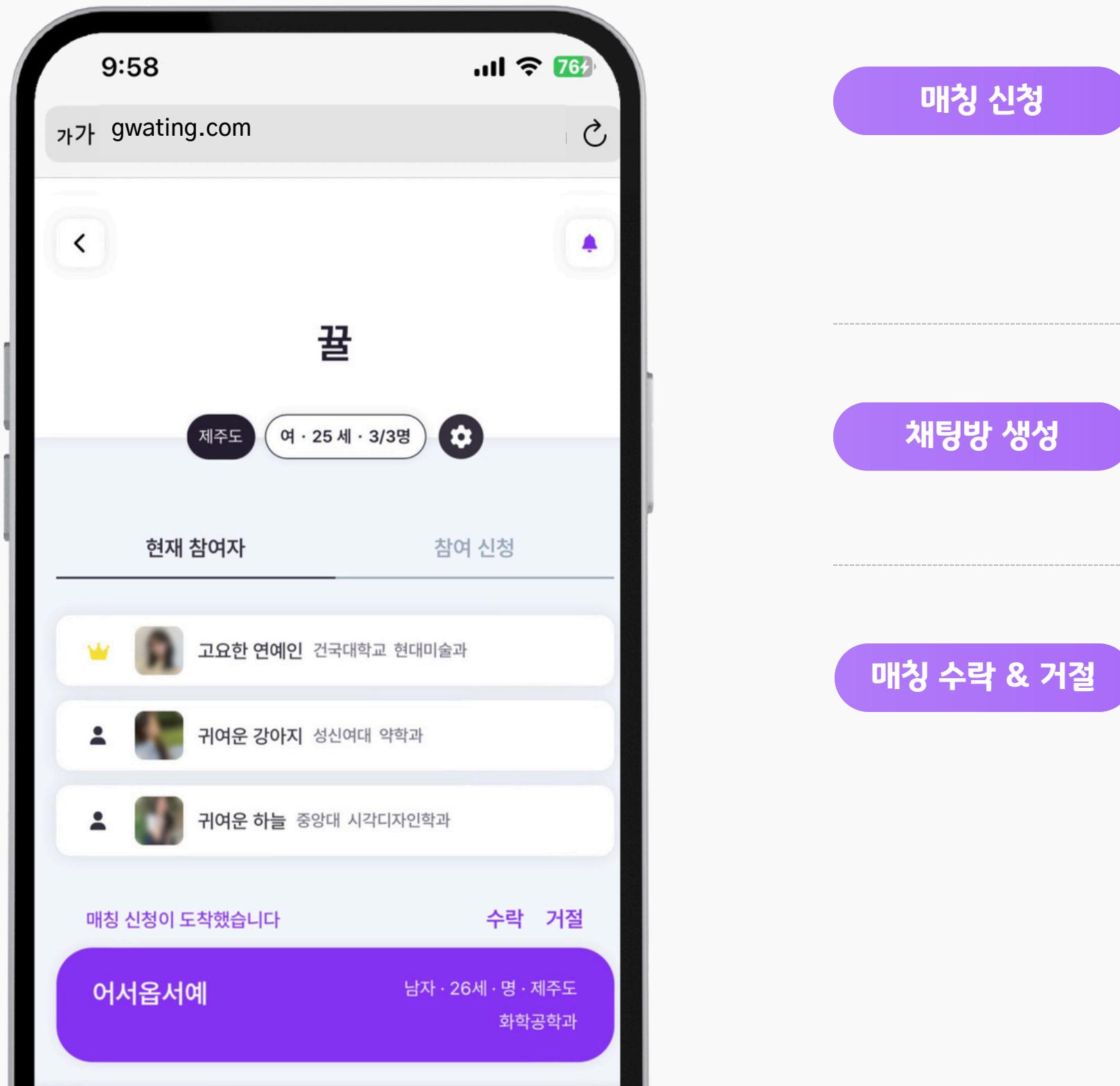
(3:3 / 4:4 / 5:5)선택 가능

그룹 생성 시 초대코드 함께 생성

미팅을 함께 하고 싶은 친구에게 초대코드 발송

주최자한테 초대 받은 사람이 참여 링크로 이동 시
미팅 그룹에 참여 신청이 완료

기능 소개 | 그룹 매칭

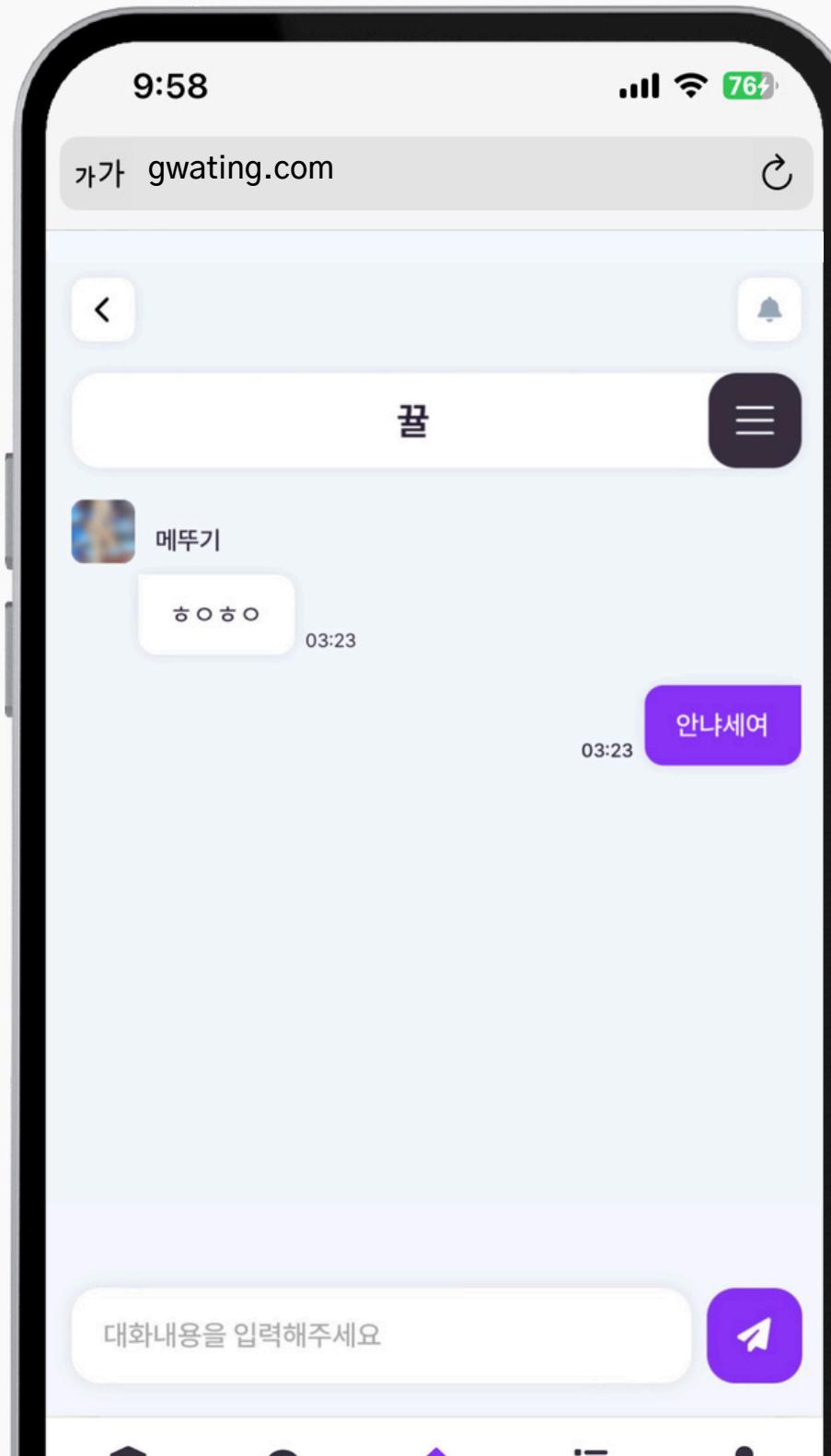


신청자그룹에서 주최자그룹에 매칭 신청
희망지역 / 참여인원 일치하고
이성 그룹일 때 신청 가능
이외의 경우 신청 불가

주최자 그룹에 매칭신청 수락시
채팅방 생성 및 입장

주최자 그룹에 매칭신청 수락 또는 거절
주최자 그룹에서 수락시 채팅방 생성
수락시 나머지 신청 그룹 거절 처리

기능 소개 | 채팅



채팅 참여 유저

상단 메뉴에 채팅 참여 유저 버튼 클릭 시 채팅에 참여중인
유저 정보 조회 가능
그룹별 유저 조회 가능

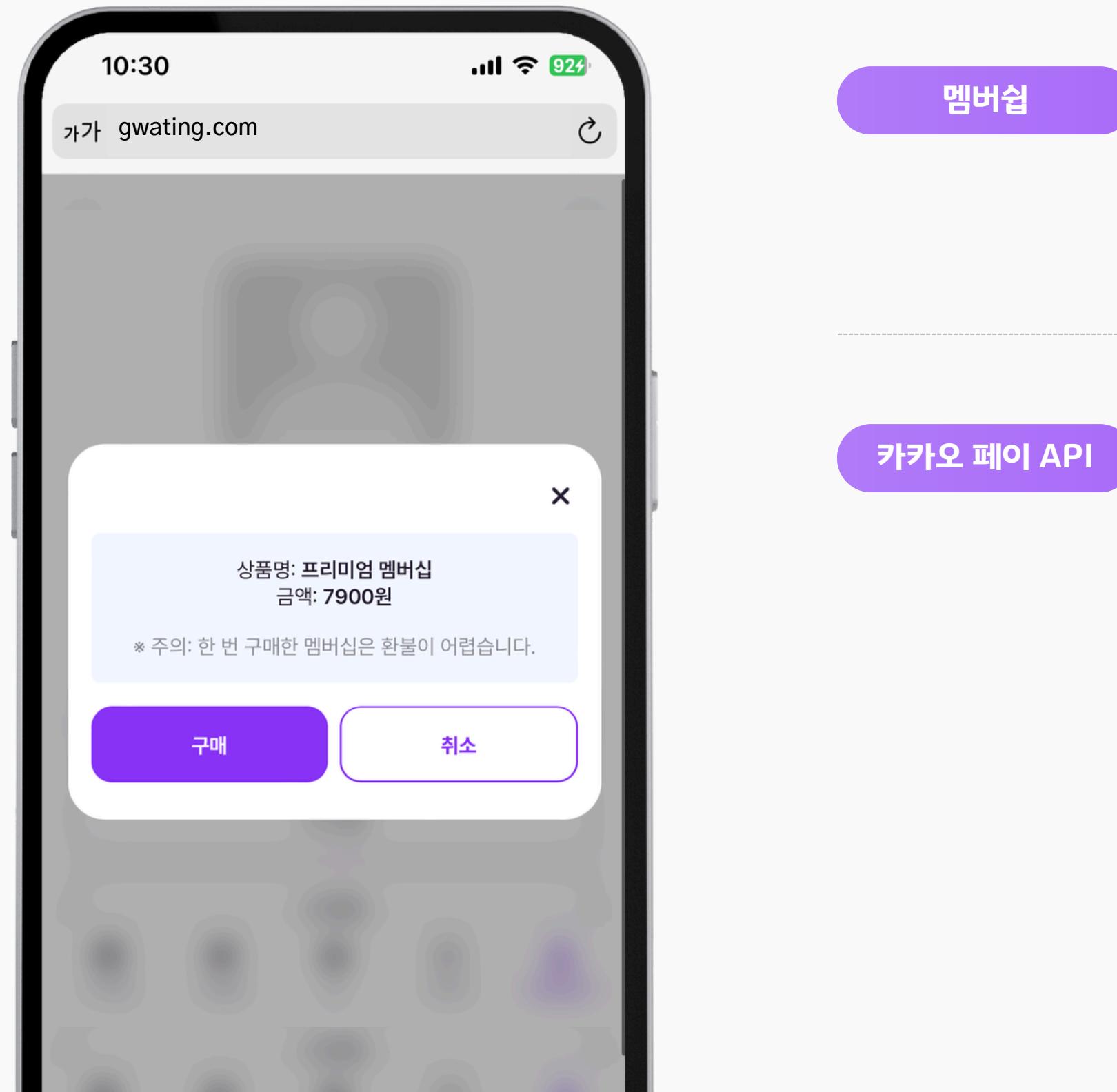
채팅 삭제

상단 메뉴에 채팅 삭제 버튼 클릭 시 채팅 삭제 가능
채팅 삭제 시 채팅이 종료되었다는 메세지 출력
채팅 삭제시 참여하고 있는 그룹도 같이 삭제

채팅 메세지

채팅방 입장 시 해당 채팅방의 채팅 렌더링
실시간 채팅 가능

기능 소개 | 멤버쉽



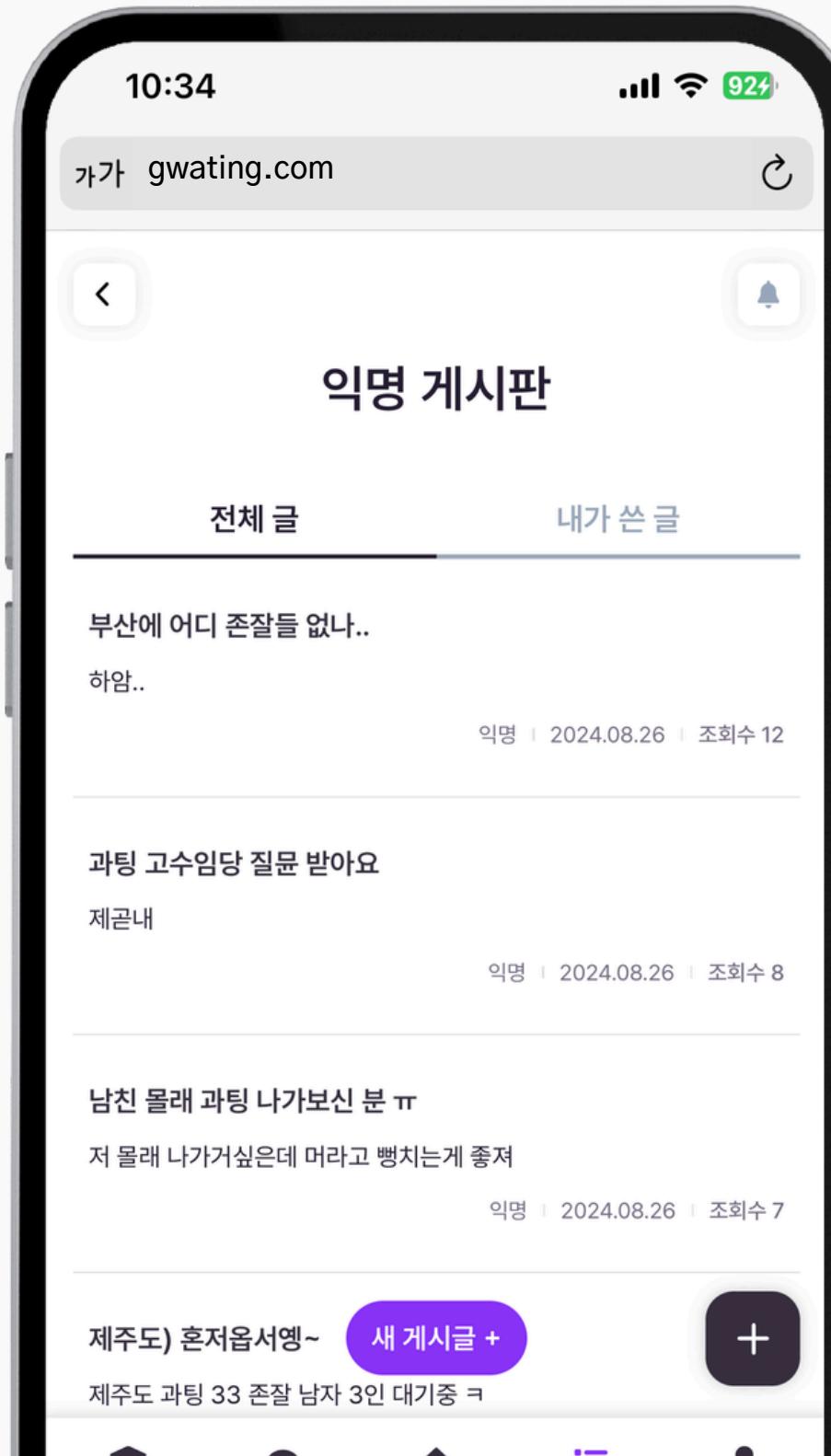
GENERAL 등급

PREMIUM 등급

- 프로필 열람, 그룹 생성 수 증가
- HOST로 있는 그룹에 들어온 매칭 요청을 조회 가능

카카오페이 API를 활용하여 모바일 결제 구현
상대방 프로필을 누를 시 또는 마이페이지에서 결제 가능

기능 소개 | 익명게시판

**게시글 생성**

새 게시글 등록 버튼을 통해 작성 폼으로 이동
제목과 내용을 입력하고 확인 버튼을 클릭하면 새 게시글 등록

게시판 열람

익명으로 게시된 자유주제 게시글을 열람 가능

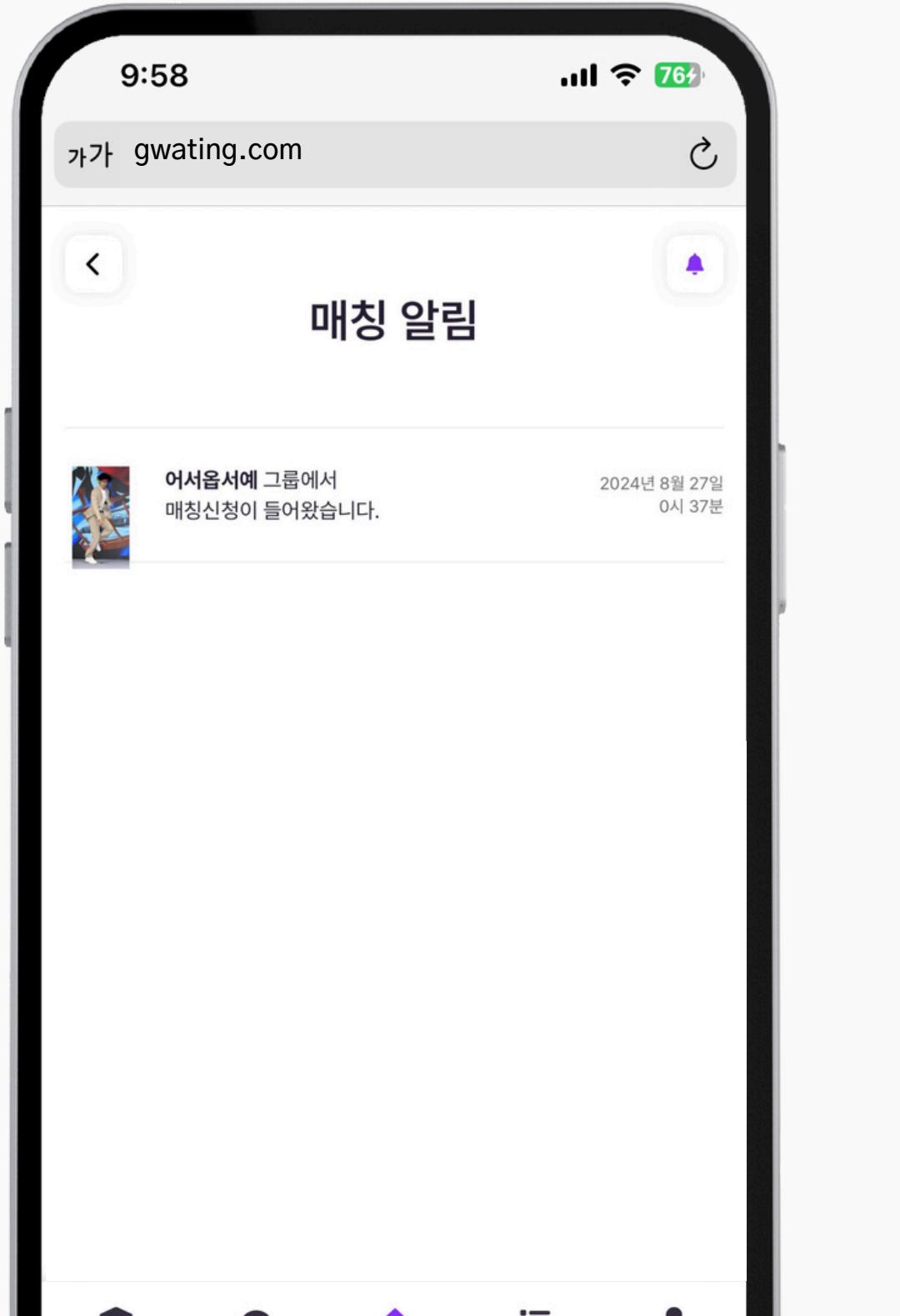
댓글

각 게시물에 익명으로 댓글 작성 가능
내가 작성한 댓글이라면 삭제 가능

게시글 수정

내가 쓴 글 탭을 클릭하면 사용자가 작성한 게시글
목록 열람 가능
해당 게시물들은 수정, 삭제 가능

기능 소개 | 알림



실시간 알림

알림 페이지

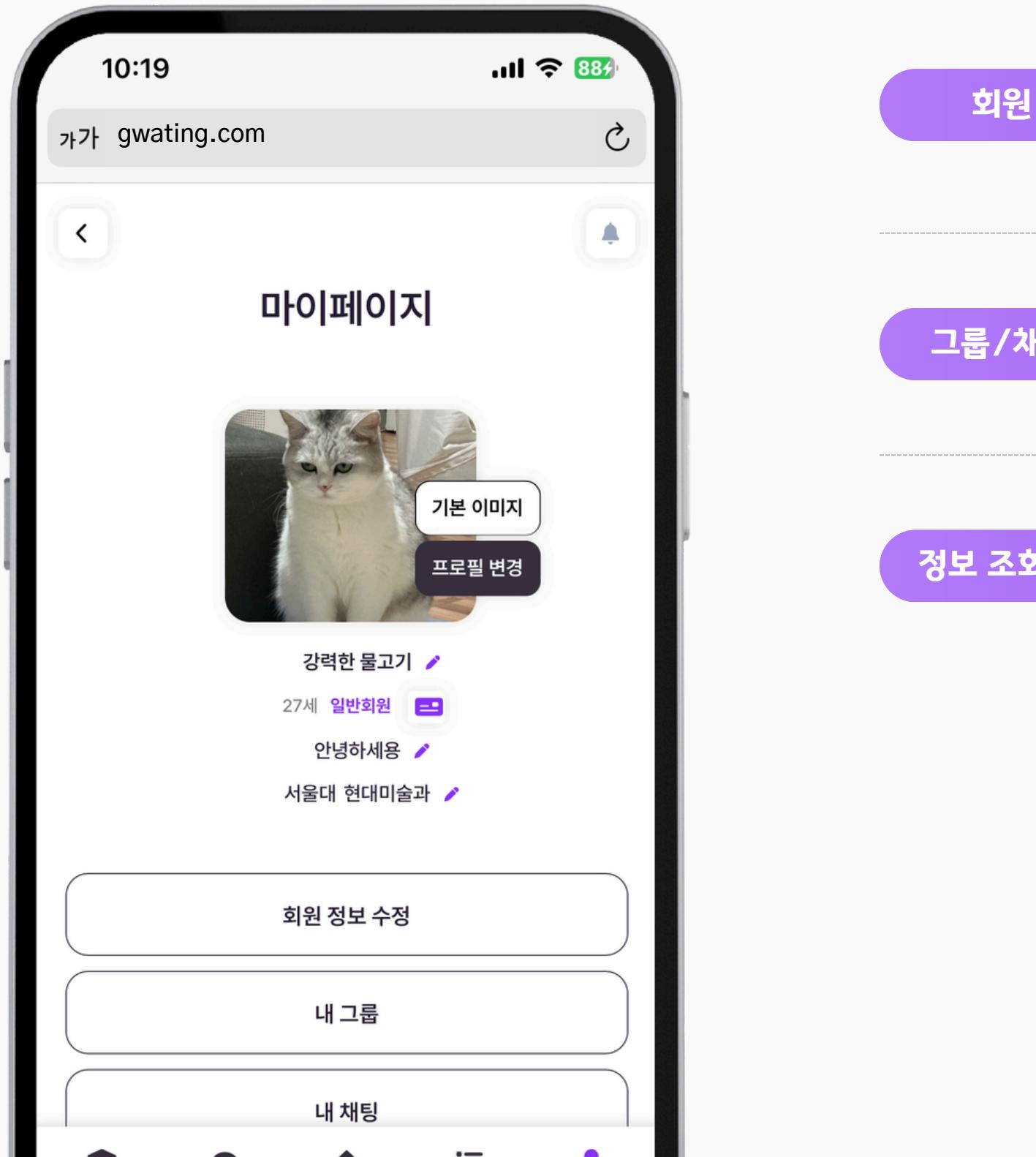
알림 목록

사용자가 HOST로 있는 그룹에 매칭 신청이 들어오면
실시간으로 알림 메세지 전송

자신이 HOST로 있는 그룹에 들어온 매칭 요청을 조회 가능

알림을 클릭하면 매칭 신청이 들어온 그룹페이지로 이동

기능 소개 | 마이페이지



이메일 확인 → 인증코드 확인 → 비밀번호
확인 절차를 거쳐 회원 탈퇴 진행

내가 속한 그룹 및 내가 속한 채팅 조회

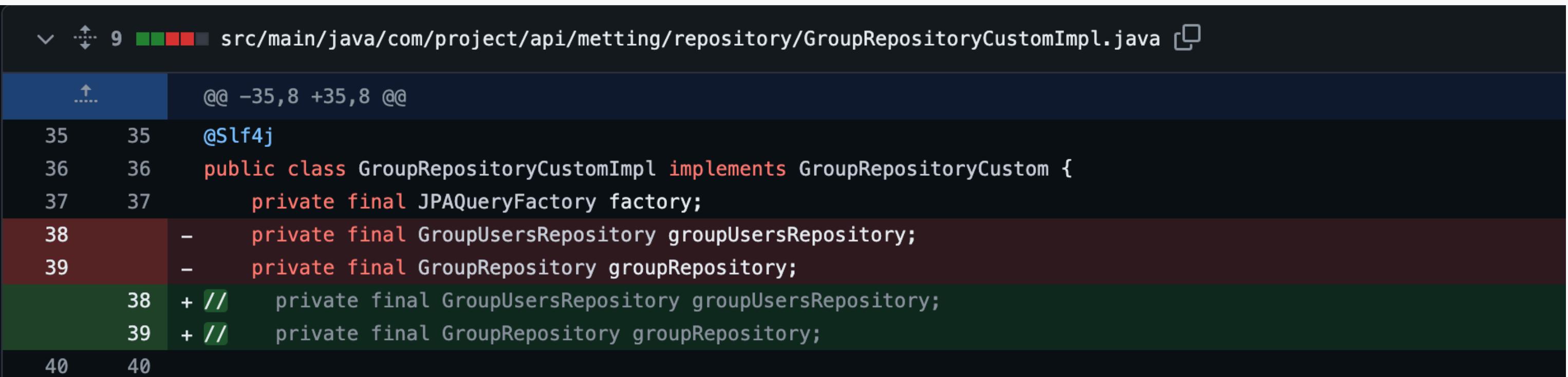
정보 조회 및 변경

기본 정보 조회 및 변경
(닉네임 / 자기소개 / 전공 / 나이 / 멤버십 등급)
기본 프로필 이미지 설정
원하는 프로필 이미지 설정
비밀번호 & 연락처 변경
연락처 중복 확인

트러블슈팅

문제발생

Repository 클래스 내에서 다른 엔터티의 Repository를 호출하는 과정에서 순환 의존성이 발생하여 애플리케이션 구동 시점에 BeanCurrentlyInCreationException 또는 StackOverflowError 등의 예외가 발생하고, 애플리케이션이 동작하지 않는 문제가 발생했다.



```
diff --git a/src/main/java/com/project/api/metting/repository/GroupRepositoryCustomImpl.java b/src/main/java/com/project/api/metting/repository/GroupRepositoryCustomImpl.java
--- a/src/main/java/com/project/api/metting/repository/GroupRepositoryCustomImpl.java
+++ b/src/main/java/com/project/api/metting/repository/GroupRepositoryCustomImpl.java
@@ -35,8 +35,8 @@
 35 35     @Slf4j
 36 36     public class GroupRepositoryCustomImpl implements GroupRepositoryCustom {
 37 37         private final JPAQueryFactory factory;
 38 38         -     private final GroupUsersRepository groupUsersRepository;
 39 39         -     private final GroupRepository groupRepository;
 38 38         + //     private final GroupUsersRepository groupUsersRepository;
 39 39         + //     private final GroupRepository groupRepository;
 40 40 
```

해결방법

Repository 간의 직접적인 의존성을 제거하고, 서비스 계층에서 필요한 로직을 처리하도록 리팩토링했다.
Repository는 데이터 접근에만 집중하고, 비즈니스 로직은 서비스 계층에서 처리하도록 구조를 수정했다.

트러블슈팅

문제발생

알림기능을 구현하기 위하여 Websocket을 사용하였는데, 연결한 웹소켓을 불러오는 것에 대해 문제가 생겼다.
처음엔 Props로 전달하여 관리하려고 했지만, 많은 요소들이 있어 Props로 전달하기 매우 어려운 상황에 처했다.
먼저, 처음 Websocket을 불러오기 위해 redux를 사용하여 slice형태로 저장하고 관리하려고 했지만,
redux는 동기적인 상태관리 도구이지 비동기적 요소를 관리하려면 추가적인 미들웨어를 사용해야 한다하여 이를 포기하고 다른 방법을 찾았다.

```
import { createContext } from "react";

export const MainWebSocketContext : Context<{...}> = createContext( defaultValue: {
    mainWebSocket:null,
}
);
|  
  
const mainSocket :{mainWebSocket: null} = useContext(MainWebSocketContext);  
  
return (
    <MainWebSocketContext.Provider
        value={{
            mainWebSocket: mainSocket,
        }}
    >
```

해결방법

이를 해결하기 위하여 Websocket을 Context로 관리 했다.
웹소켓이 연결되는 곳에서 연결된 웹소켓을 Context에 저장하고 provide 하고 다른 곳에서 useContext를 이용하여 연결된 웹소켓과 상호작용했다.

트러블슈팅

문제발생

회원가입시 이메일 인증코드를 입력 하기 전에 인증코드를 다시 보냈을 경우에는 메일에 오지만 인증코드를 입력 후에 다시 회원가입하여 인증코드를 보냈을 경우에는 메일로 인증코드가 오지 않는 문제가 발생했다.

```
if (exists) {
    // 사용자가 존재하고 인증이 완료되지 않았을 경우 인증 코드 재발송
    if (notFinish(email)) {
        // 중복된 유저가 존재하면 해당 유저를 삭제
        User user = userRepository.findByEmail(email).orElseThrow();
        clearAndResendVerification(email, user.getUnivName());
        log.info("Deleting existing user with email: {}", email);
        userRepository.delete(user);

        // 기존 사용자가 삭제되었으므로 새 사용자 등록
        processSignUp(email, univName);
        return false;
    }
    return true;
} else {
    processSignUp(email, univName); // 신규 사용자 등록
    return false;
}
```



해결방법

기존 중간처리 로직을 없애고 데이터베이스에서 회원가입을 진행했던 유저의 정보를 삭제하고 다시 만드는 식으로 수정했다.

트러블슈팅

문제발생

프로필 사진 변경시 기존 사진이 랜더링 되고, 변경한 사진이 업로드 되지 않는 문제

```
if (response.ok) {
  const result = await response.text();
  console.log("프로필 이미지가 업데이트되었습니다.: ", result);
  setProfileImg(result); // 서버에서 받은 이미지 URL을 상태에 저장
} else {
  throw new Error("프로필 이미지를 업데이트 하지 못했습니다.");
}
```

```
if (response.ok) {
  const imageUrl = await response.json();
  setProfileImg(imageUrl.profileImg); // 가져온 프로필 이미지를 상태에 저장
} else {
  const errorMessage = `프로필 이미지를 가져오지 못했습니다.: ${response.status} ${response.statusText}`;
  throw new Error(errorMessage);
}
} catch (error) {

  console.error("프로필 이미지를 가져오는 중에 오류가 발생했습니다.: ", error.message);
}
```

해결방법

S3에 파일을 업로드하고 업로드한 파일이 아니라 파일의 경로를 DB에 저장하기 때문에 파일의 경로를 setProfileImg(imageUrl.profileImg) 와 같이 저장하여 해결함

트러블슈팅

문제발생

도커를 통해서 성공적으로 배포를 했다고 생각해서 프론트와 백엔드를 연결시켜서 성공적으로 폐치를 보내려고 했는데 백엔드 서버 쪽에서 CORS 문제가 발생했었다. Controller 부분에서도 CrossOrigin을 적용했는데 따로 Config에서도 CrossOrigin 관련 설정을 해서 서로 충돌을 일으켜서 폐치를 못 보내는 상황이 있었다

```
//전역 크로스 오리진 설정 : 어떤 클라이언트를 허용할 것인지
@Configuration 0개의 사용위치 hun2zz

public class CrossOriginConfig implements WebMvcConfigurer {

    private String[] url = { 1개 사용 위치
        "http://localhost:3000",
        "http://localhost:3001",
        "http://localhost:3002",
        "http://gwating.com.s3-website.ap-northeast-2.amazonaws.com"
    };
}
```

해결방법

이를 해결하기 위해서 모든 컨트롤러 부분에서 따로 CrossOrigin 아노테이션을 지우고 따로 config에서 CORS 관련 설정을 따로 해주었다.

트러블슈팅

문제발생

다중 필터링 기능 구현 시, 각 필터의 상태를 useState로 관리하다 보니 코드가 복잡해지고 유지보수가 어려워졌습니다.

또한, 초기에는 필터 상태를 서버로 전달하기 위해 POST 요청을 사용했지만, 이 과정에서 필터 상태가 올바르게 전달되지 않는 문제가 발생했습니다.

```
14 + const [searchParams, setSearchParams] = useSearchParams();
15 + const pageNo = Number(searchParams.get("pageNo")) ?? 1;
16 + const gender = searchParams.get("gender");
17 + const region = searchParams.get("region");
18 + const personnel = searchParams.get("personnel");
```

```
46 - const response = await fetch(`http://localhost:8253/main/${pageNo}`, {
42 + // 동적 url 만들기
43 + let url = `http://localhost:8253/main?pageNo=${pageNo}`;
44 + if (gender) {
45 +   url += `&gender=${gender}`;
46 + }
47 + if (personnel) {
48 +   url += `&personnel=${personnel}`;
49 + }
50 + if (region) {
51 +   url += `&region=${region}`;
52 + }
```



해결방법

필터 상태 전달 방식을 POST에서 GET으로 전환하여 구현하였으며, 각 필터의 상태 관리를 useState 대신 useSearchParams를 활용하여 개선했습니다.

이를 통해, 필터 조건이 변경될 때마다 동적으로 URL을 생성하여 서버에 요청을 보내는 방식으로 로직을 재구성했습니다.

향후
업데이트



향후 업데이트

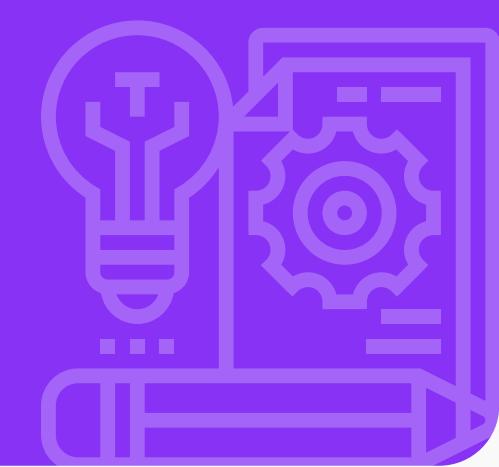
채팅

- 이미지 전송 기능
- 메세지 공지사항 등록 기능
- 채팅 메세지 알림 기능



익명 게시판

- 게시글 좋아요 기능
- 해시태그 기능
- 대댓글 기능



마이 페이지

- 사용자 신고/차단 기능
- 사용자 연락처 동기화
- 멤버십 프리미엄 결제 혜택 개선



알림

- 매칭 신청 알림 뿐만 아니라
- 신청 수락, 거절에 대한
- 실시간 알림 처리



한 줄 소감



진상훈

백, 프론트, 배포 전부 다양하게 할 수 있었던 프로젝트여서 나에게 의미가 깊었던 프로젝트인 거 같고 재밌었다. 즐겁게 개발하자! 🎉

오전 11:03

김은지



이번 프로젝트를 통해 이전 프로젝트보다 문제를 맞닥뜨릴 때 해결할 수 있는 능력을 향상시킬 수 있었다. 예상하지 못한 오류와 장애를 만나면서 디버깅하는 방법들을 깨우쳤다. 팀원들을 통해 협업 방법에 대해 배웠다. 서로의 의견을 존중하고 조율하면서 하나씩 해결해 나가는 모습들이 성공적으로 프로젝트를 마무리할 수 있었다고 생각이 들었다. 고마워! ❤️



김요한

오전 11:05

백엔드 로직 구현을 완료하고 렌더링 부분을 테스트 코드를 완벽하게 진행하지 않고 넘어갔을 때 많은 문제가 발생하여 테스트를 꼼꼼하게 진행해야 한다는 것을 알게 된 프로젝트였습니다. 또한 디자인부분에 대해 팀원들에게 많은 것을 배울 수 있었습니다. 다들 감사했습니다~ ^__^💡

오전 11:08

한 줄 소감



문지은

사용자의 행위에 따라 동시다발적인 요청이 들어갈 수 있기 때문에
상황별 유스케이스 구체화의 중요성을 깨달았다. 화이팅! 😊

오전 11:20

박진우



팀원과의 소통의 원활해서 좋았습니다. 다만, 로컬에서와 다르게 배포환경에서
오류가 많이 난다는 것을 알았고 다음부턴 배포를 조금 더 일찍하여 테스트 기간을
늘려야겠다고 생각했습니다. 즐겨~ 💪



이예진

오전 11:21

이번 프로젝트에서 디자인부터 코딩까지 체계적으로 진행할 수 있어서 정말 좋았고,
어려운 부분이 있을 때마다 서로 도와주고 배워가며 실력이 한층 더 향상된 것 같아
매우 뿌듯합니다. 함께 성장할 수 있는 좋은 경험이었습니다! 😊

오전 11:22

감사합니다

3on3 채팅방 나가기

질문

