

최종 보고서

객 체 지 향 패 러 다 임

일육이공

목차

- 01** 개요
- 02** Use-case 설계
- 03** Sequence diagram
- 04** class diagram
- 05** DB 설계
- 06** GUI설계
- 07** Requirements vs implementation

01 개요

가톨릭대학교 밥 약속 채팅 어플리케이션

밥 같이 먹을 사람??



01 개요

Chat

음식점 기반의 채팅방 생성 및 참여



역곡(가톨릭대) 주변 음식점

Talk

채팅을 통한 밥 약속 잡기



기존 채팅 앱과 유사한 채팅 기능

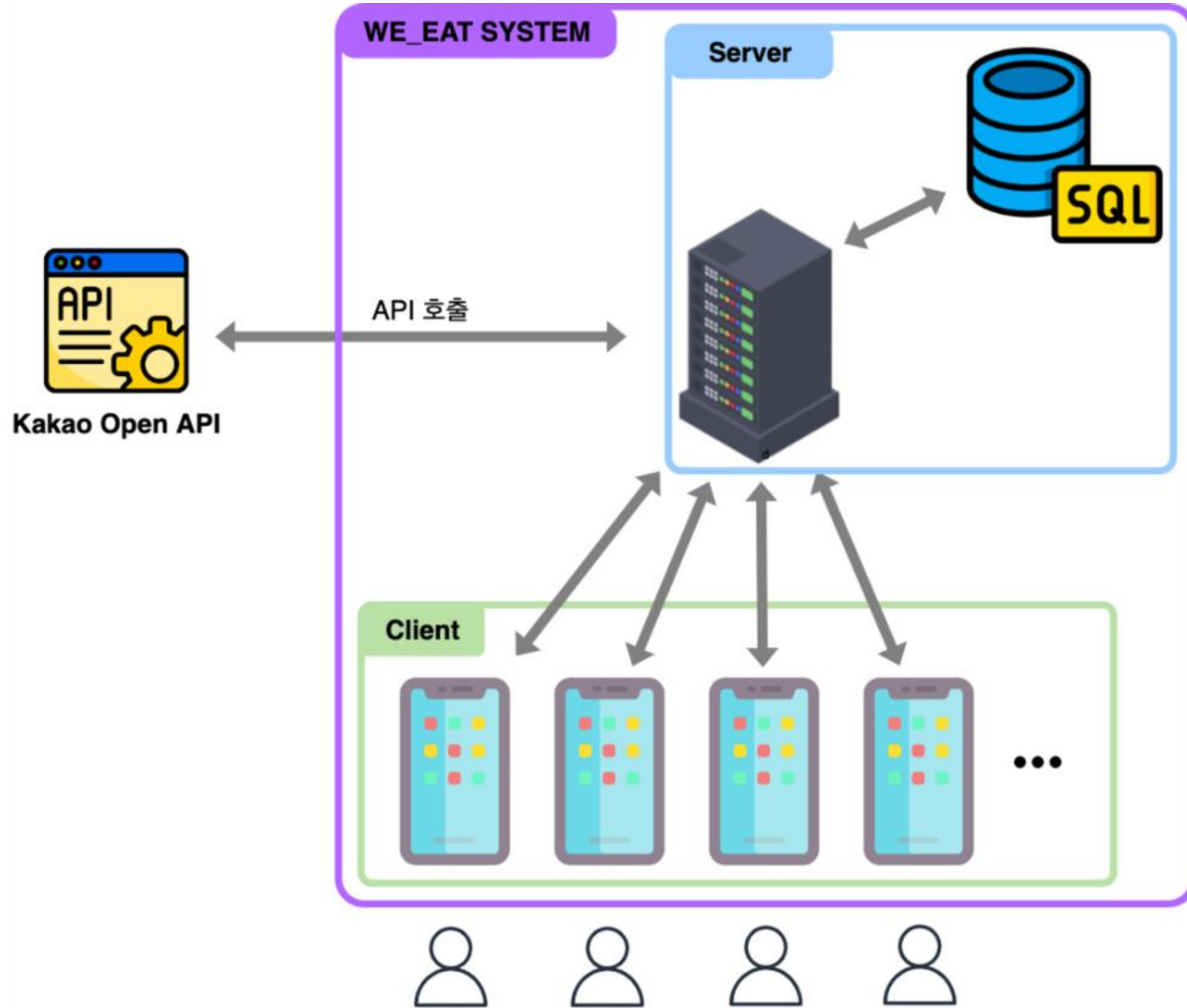
Profile

프로필 확인 및 회원가입



학번, 학과를 포함한 회원가입
사용자 프로필 확인

01 개요



Server ↔ Client

클라이언트간의 채팅을 위한
소켓통신 및 채팅시스템을 위한 API 지원

Server ↔ Kakao API

음식점 정보를 얻기 위해
외부 Kakao open API 호출

Server ↔ DB

채팅시스템에 필요한 유저 및 채팅 정보
를 저장하기 위한 데이터베이스 지원

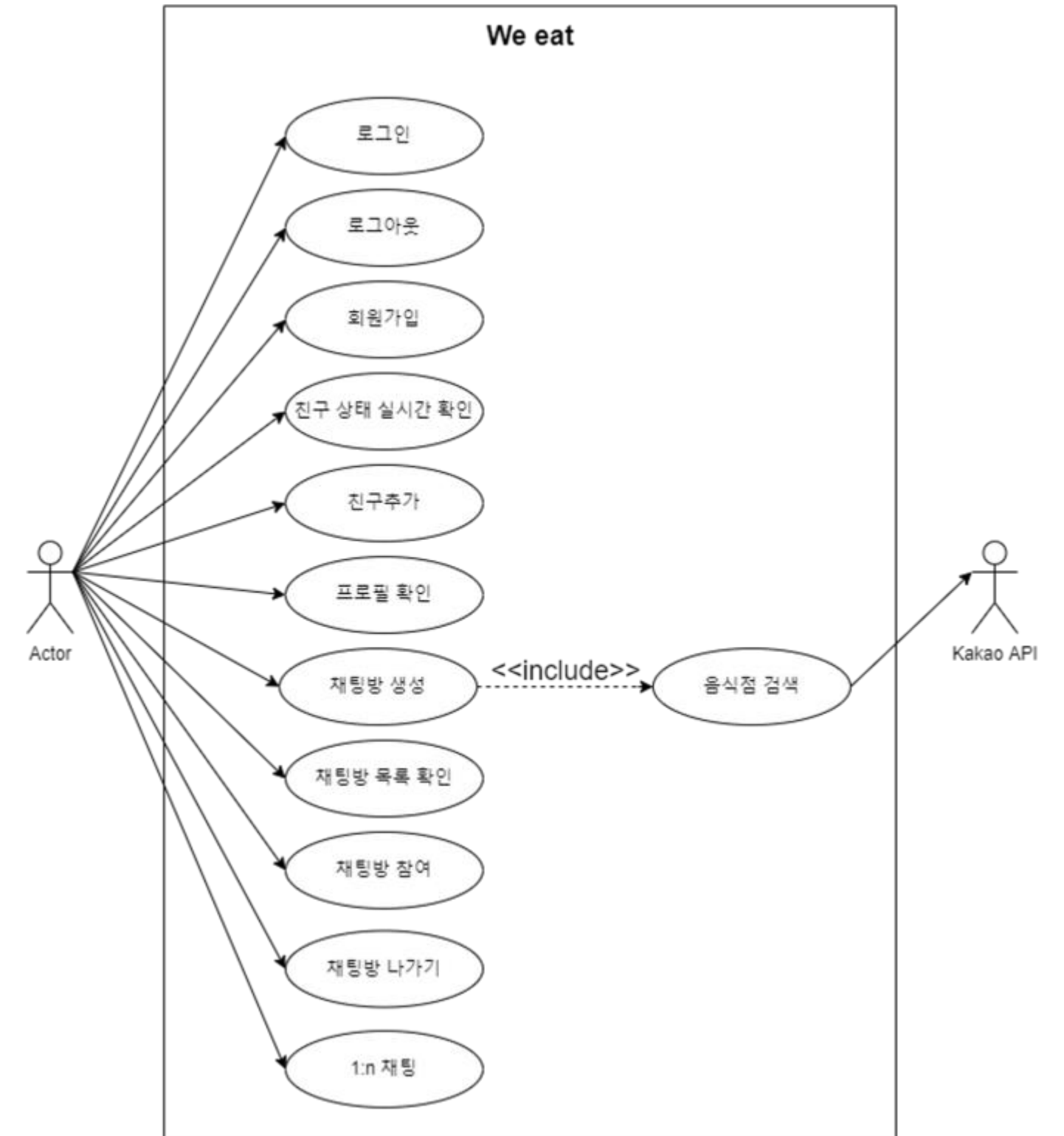
02 Use-case 설계 01. 다이어그램

액터 목록

액터	설명
User	프로그램을 사용하는 사람
Kakao API	키워드로 장소를 검색하는 API 시스템

유즈케이스 목록

유즈케이스 ID	유즈케이스명	설명
WE_SignUp	회원가입	사용자의 정보 등록
WE_SignIn	로그인	아이디와 비밀번호를 입력하여 접속
WE_SignOut	로그아웃	사용자의 접속 해제
WE_CheckStatus	친구상태실시간확인	사용자의 친구 온·오프라인 상태 표시
WE_AddFriend	친구추가	메신저에 새로운 친구를 등록
WE_CreateChatRoom	채팅방 생성	새로운 채팅방 생성
WE_CheckChatRoomList	채팅방 목록 확인	전체 채팅방 목록 확인
WE_JoinChatRoom	채팅방 참여	채팅방 목록에서 원하는 채팅방 참여
WE_ExitChatRoom	채팅방 나가기	현재 채팅방에서 나가기
WE_Chatting	1:n 채팅	텍스트 입력창에서 메시지 송수신(n명의 사용자)
WE_CheckProfile	프로필확인	프로필을 누르면 사용자의 이름 학번 학과 확인
WE_SearchRestaurant	음식점 검색	음식을 검색하면 역곡주변의 음식점들을 보여줌



회원가입 (WE_SignUp)

유즈케이스명	회원가입	
설명	사용자의 정보를 등록한다.	
관련 액터	User	
기본 흐름	액터	시스템
	1. 사용자가 회원가입 화면에서 아이디, 비밀번호, 이름, 학번, 전공을 입력하고 회원가입 버튼을 누른다.	
		2. 입력받은 정보가 올바른지 확인한다. (아이디 학번 중복확인)
		3. 입력받은 정보를 DB에 추가한다.
		4. 홈 화면으로 이동한다.
5. 유즈케이스를 종료한다.		

회원가입 (WE_SignUp)

유즈케이스명	회원가입	
	액터	시스템
예외 흐름 1	1. 아이디,비밀번호,이름,학번,전공 중 하나라도 입력하지 않고 회원가입 버튼을 누른다.	
		2. 모든 정보가 입력되지 않았다는 메시지를 출력한다.
	3. 유즈케이스를 종료한다.	
예외 흐름 2	1. DB에 이미 존재하는 아이디 또는 학번을 적고 회원가입 버튼을 누른다.	
		2. 이미 존재하는 아이디라는 메시지를 출력한다.
	3. 유즈케이스를 종료한다.	

로그인 (WE_SignIn)

유즈케이스명	로그인	
설명	아이디와 비밀번호를 입력하여 접속한다.	
관련 액터	User	
기본 흐름	액터	시스템
	1. 사용자가 로그인 화면에서 아이디, 비밀번호를 입력하고 로그인 버튼을 누른다.	
		2. 입력받은 정보를 DB로 보내 아이디 비밀번호가 일치하는지 확인한다.
		3. DB에 접근하여 사용자 상태를 온라인으로 변경한다.
		4. 사용자 정보를 포함한 홈 화면으로 이동한다.
5. 유즈케이스를 종료한다.		

로그인 (WE_SignIn)

유즈케이스명	로그인	
	액터	시스템
예외 흐름 1	1. 아이디 또는 비밀번호를 입력하지 않고 로그인 버튼을 누른다. 3. 유즈케이스를 종료한다.	2. 모든 정보가 입력되지 않았다는 메시지를 출력한다.
예외 흐름 2	1. 잘못된 아이디나 비밀번호를 입력하고 로그인 버튼을 누른다. 3. 유즈케이스를 종료한다.	2. 존재하지 않은 아이디이거나 비밀번호가 틀렸다는 메시지를 출력한다.

로그아웃 (WE_SignOut)

유즈케이스명	로그아웃	
설명	사용자의 접속을 해제한다.	
관련 액터	User	
기본 흐름	액터	시스템
	1. 사용자가 홈 화면에서 로그아웃 버튼을 누른다.	
		2. DB에 접근하여 사용자 상태를 오프라인으로 변경한다.
		3. 로그인 화면으로 이동한다.
	4. 유즈케이스를 종료한다.	

친구 상태 실시간 확인 (WE_CheckStatus)

유즈케이스명	친구 상태 실시간 확인	
설명	사용자 친구의 온·오프라인 상태를 확인한다.	
관련 액터	User	
기본 흐름	액터	시스템
	1. 사용자가 홈 화면에 접근한다.	2. DB에 접근하여 사용자의 친구를 확인하고 친구들의 온·오프라인 상태를 리스트 형태로 보여준다.
	3. 유즈케이스를 종료한다.	

친구 추가 (WE_AddFriend)

유즈케이스명	친구추가	
설명	메신저에 새로운 친구를 등록한다.	
관련 액터	User	
기본 흐름	액터	시스템
	1. 홈 화면에서 친구추가버튼을 누른다.	2. 친구추가 화면을 보여준다.
	3. 사용자 이름을 입력한다.	4. DB에 접근해서 일치하는 이름 목록을 리스트 형태로 보여준다.
	5. 사용자가 추가하고 싶은 친구를 클릭한다.	7. DB에 접근해서 사용자와 친구와의 관계를 추가한다.
	6. 친구추가 버튼을 누른다.	
	8. 유즈케이스를 종료한다.	
예외 흐름	1. DB에 없거나 이미 친구인 사용자 이름을 입력한다.	2. '이미 친구이거나 해당 사용자가 없습니다'라는 메시지를 출력한다.
	3. 유즈케이스를 종료한다.	

채팅방 생성 (WE_CreateChatRoom)

유즈케이스명	채팅방 생성	
설명	사용자가 새로운 채팅방을 생성한다.	
관련 액터	User	
기본 흐름	액터	시스템
	1. 전체 채팅방 목록 화면 하단에 위치한 '+'버튼을 클릭한다.	
		2. 채팅방 생성 화면을 사용자에게 보여준다.
	3. 생성할 채팅방 이름을 입력한다.	
	4. '음식점 검색' 기본 흐름으로 이동한다.	
	5. '친구 추가' 텍스트 오른쪽에 위치한 '+'버튼을 클릭한다.	
		6. 사용자의 친구목록을 리스트 형태로 보여준다.
	7. 리스트에 있는 친구(들)을 선택하고 '친구 추가'버튼을 클릭한다.	
	8. '채팅방 생성'버튼을 클릭한다.	
		9. 사용자가 선택한 음식점, 친구 데이터를 기반으로 채팅방을 생성한다.
		10. 채팅방 파일을 생성한다.
		11. 채팅방을 생성한 사용자, 초대받은 친구(들), 채팅방파일경로를 DB에 기록한다.
12. 유즈케이스를 종료한다.		

채팅방 생성 (WE_CreateChatRoom)

유즈케이스명	채팅방 생성	
설명	사용자가 새로운 채팅방을 생성한다.	
관련 액터	User	
예외 흐름	액터	시스템
	1. 채팅방 이름이나 음식점을 입력하지 않고 채팅방 생성 버튼을 클릭한다.	2. 모든 정보가 입력되지 않았다는 메시지를 출력한다.
	3. 유즈케이스를 종료한다.	

채팅방 목록 확인 (WE_CheckChatRoomList)

유즈케이스명	채팅방 목록 확인	
설명	생성되어있는 채팅방 목록을 확인한다.	
관련 액터	User	
기본 흐름1	액터	시스템
	1. 사용자가 전체 채팅방 목록 화면에 접근한다.	2. DB에 접근하여 현재 생성된 채팅방(채팅방 이름, 음식점, 생성시간) 목록을 출력한다.
	3. 유즈케이스를 종료한다.	
기본 흐름2	1. 사용자가 홈 화면에 접근한다.	2. DB에 접근하여 사용자가 참여중인 채팅방 목록(채팅방 이름, 음식점, 생성시간)을 출력한다.
	3. 유즈케이스를 종료한다.	

채팅방 참여(WE_JoinChatRoom)

유즈케이스명	채팅방 참여	
설명	채팅방 목록에서 사용자가 원하는 채팅방에 참여한다.	
관련 액터	User	
기본 흐름 1	액터	시스템
	1. 참여 중이지 않은 채팅방을 클릭한다.	2. 선택한 채팅방에 사용자가 포함되어 있는지 DB에서 확인한다.
		3. 포함되어 있지 않으면 사용자가 채팅방에 들어갔음을 DB에 기록한다.
		4. 사용자가 선택한 채팅방화면으로 이동한다.
		5. 채팅방 파일에 채팅방에 사용자가 입장했다고 기록한다.
		6. 채팅방 파일에 담긴 대화내용(채팅방 입출입 정보포함)을 불러온다.
		7. 대화내용을 출력한다.
	8. 유즈케이스를 종료한다.	

채팅방 참여(WE_JoinChatRoom)

유즈케이스명	채팅방 참여	
설명	채팅방 목록에서 사용자가 원하는 채팅방에 참여한다.	
관련 액터	User	
기본 흐름 2	액터	시스템
	1. 참여 중인 채팅방을 클릭한다.	2. 선택한 채팅방에 사용자가 포함되어 있는지 DB에서 확인한다.
		3. 포함되어 있으면 채팅방 화면으로 이동한다.
		4. 채팅방 파일에 담긴 대화내용(채팅방 입·출입 정보 포함)을 불러온다.
		5. 대화내용을 출력한다.
	6. 유즈케이스를 종료한다.	

1:N 채팅 (WE_Chatting)

유즈케이스명	1:n 채팅	
설명	텍스트 입력창에서 메시지 송수신(n명의 사용자)	
관련 액터	User	
	액터	시스템
기본 흐름	1. 사용자가 텍스트 입력창에 메시지를 입력하고 전송버튼을 누른다.	2. 메시지를 채팅방에 들어와 있는 사람들(메세지를 작성한 사용자포함)에게 보내고 출력한다.
	3. 유즈케이스를 종료한다.	
예외 흐름 1	1. 500자 이상의 메시지를 입력한다.	2. 입력 가능한 글자 수를 초과했다는 메시지를 출력한다.
	3. 유즈케이스를 종료한다.	
예외 흐름 2	1. 사용자가 텍스트 입력창에 아무것도 입력하지 않고 전송버튼을 누른다.	2. 메시지를 입력하라는 메시지를 출력한다.
	3. 유즈케이스를 종료한다.	

프로필 확인 (WE_CheckProfile)

유즈케이스명	프로필 확인	
설명	프로필을 누르면 상대방의 정보를 확인할 수 있다.	
관련 액터	User	
기본 흐름	액터	시스템
	1. 사용자가 궁금한 사람의 프로필을 클릭한다.	2. DB에 접근하여 클릭한 사람의 이름, 학과, 학번을 출력한다.
	3. 유즈케이스를 종료한다.	

유즈케이스명	음식점 검색		
설명	사용자가 음식을 검색하면 역곡주변의 음식점을 보여준다.		
관련 액터	User, Kakao api		
기본 흐름	User	Kakao api	시스템
	1. 음식점 입력 텍스트 필드에 음식 키워드를 입력한다. ex) 치킨		
			2. 사용자가 입력한 키워드 앞에 '역곡'을 추가하고 API를 호출한다.
		3. 역곡 음식점 리스트를 시스템에 제공한다.	
			4. 역곡 음식점 리스트를 받아서 사용자에게 리스트 형태로 보여준다.
	5. 리스트에 있는 음식점 중 한 곳을 선택한다. 6. 유즈케이스를 종료한다.		
예외 흐름1		1. 시스템에 제공할 음식점이 존재하지 않는다.	2. 채팅방을 생성하는 사용자에게 음식점이 없다고 알린다.
	3. 유즈케이스를 종료한다.		
예외 흐름2	1. 음식 이름을 입력하지 않는다.		2. '음식 이름을 입력하세요'라는 메시지를 출력한다.
	3. 유즈케이스를 종료한다.		

채팅방 나가기 (WE_ExitChatRoom)

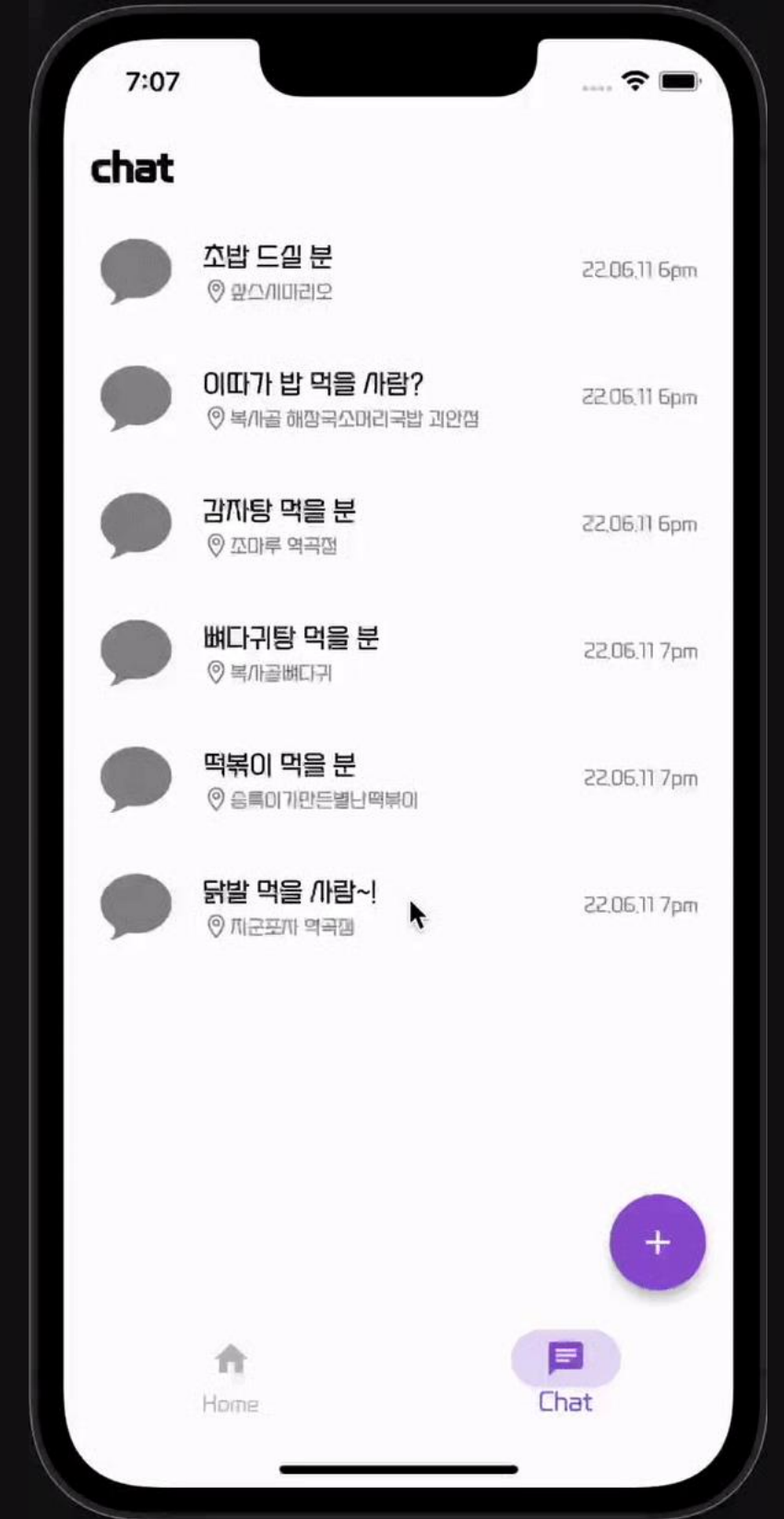
유즈케이스명	채팅방 나가기	
설명	현재 사용자가 있는 채팅방을 나간다.	
관련 액터	User	
기본 흐름1	User	시스템
	<ol style="list-style-type: none"> 1. 채팅방 화면에서 채팅방 나가기 버튼을 클릭한다. 	<ol style="list-style-type: none"> 2. DB에 접근하여 채팅방의 인원 수를 파악한다. 3. 채팅방의 인원 수가 2명 이상이면 DB에서 채팅방과 유저의 관계를 제거한다. 4. 채팅방 파일에 사용자가 나갔음을 기록한다. 5. 채팅방에 사용자가 나갔음을 출력한다. 6. 이전 화면을 보여준다.
	7. 유즈케이스를 종료한다.	

채팅방 나가기 (WE_ExitChatRoom)

유즈케이스명	채팅방 나가기	
설명	현재 사용자가 있는 채팅방을 나간다.	
관련 액터	User	
기본 흐름2	User	시스템
	1. 채팅방 화면에서 채팅방 나가기 버튼을 클릭한다.	2. DB에 접근하여 채팅방의 인원 수를 파악한다.
		3. 채팅방의 인원 수가 1명이면 해당 채팅방을 DB에서 제거한다.
		4. 이전 화면을 보여준다.
	5. 유즈케이스를 종료한다.	

1:N 채팅

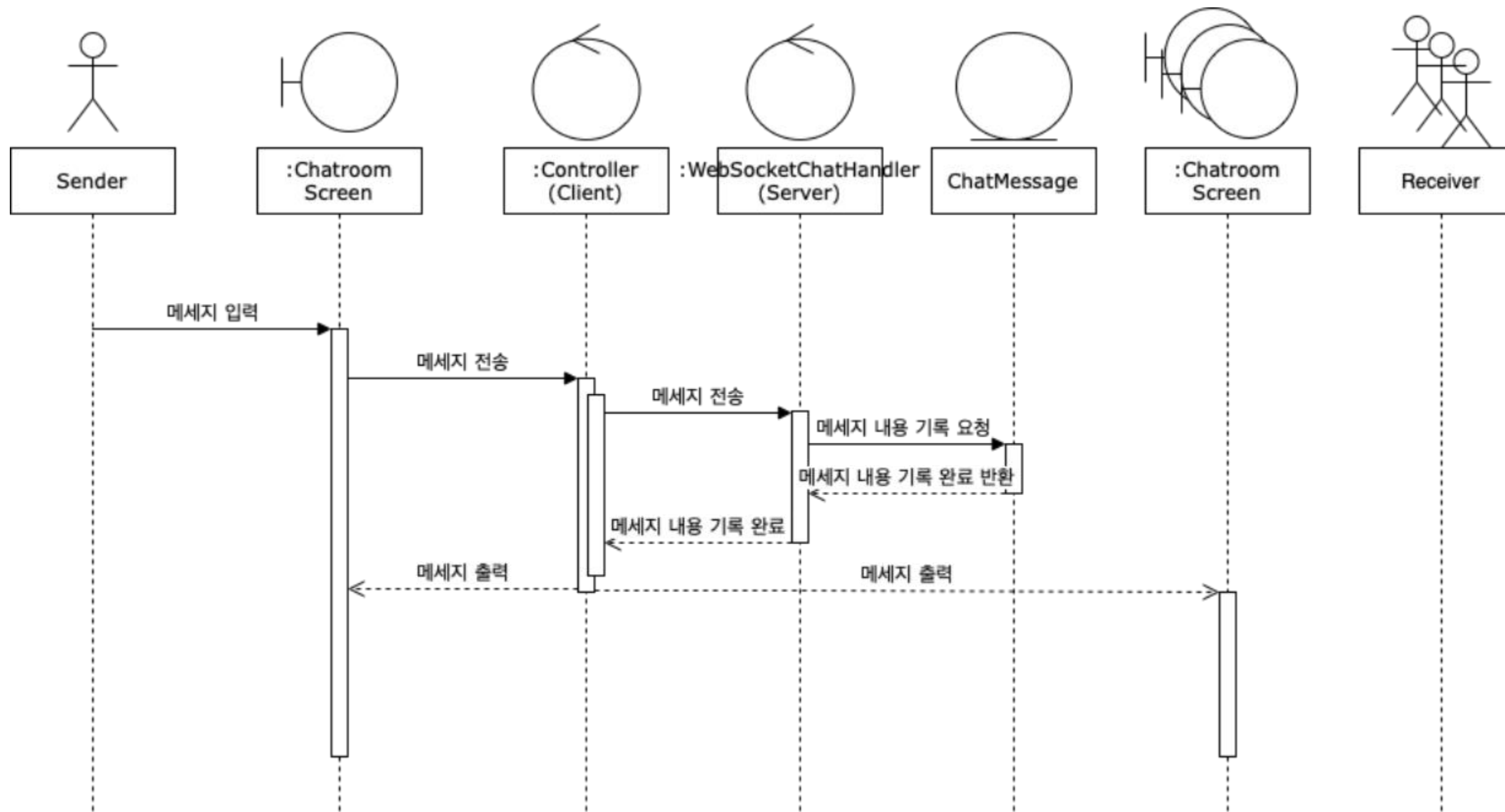
채팅방에 참여하여 N명의 사람들과 채팅합니다.



03 Sequence diagram 01. 다이어그램

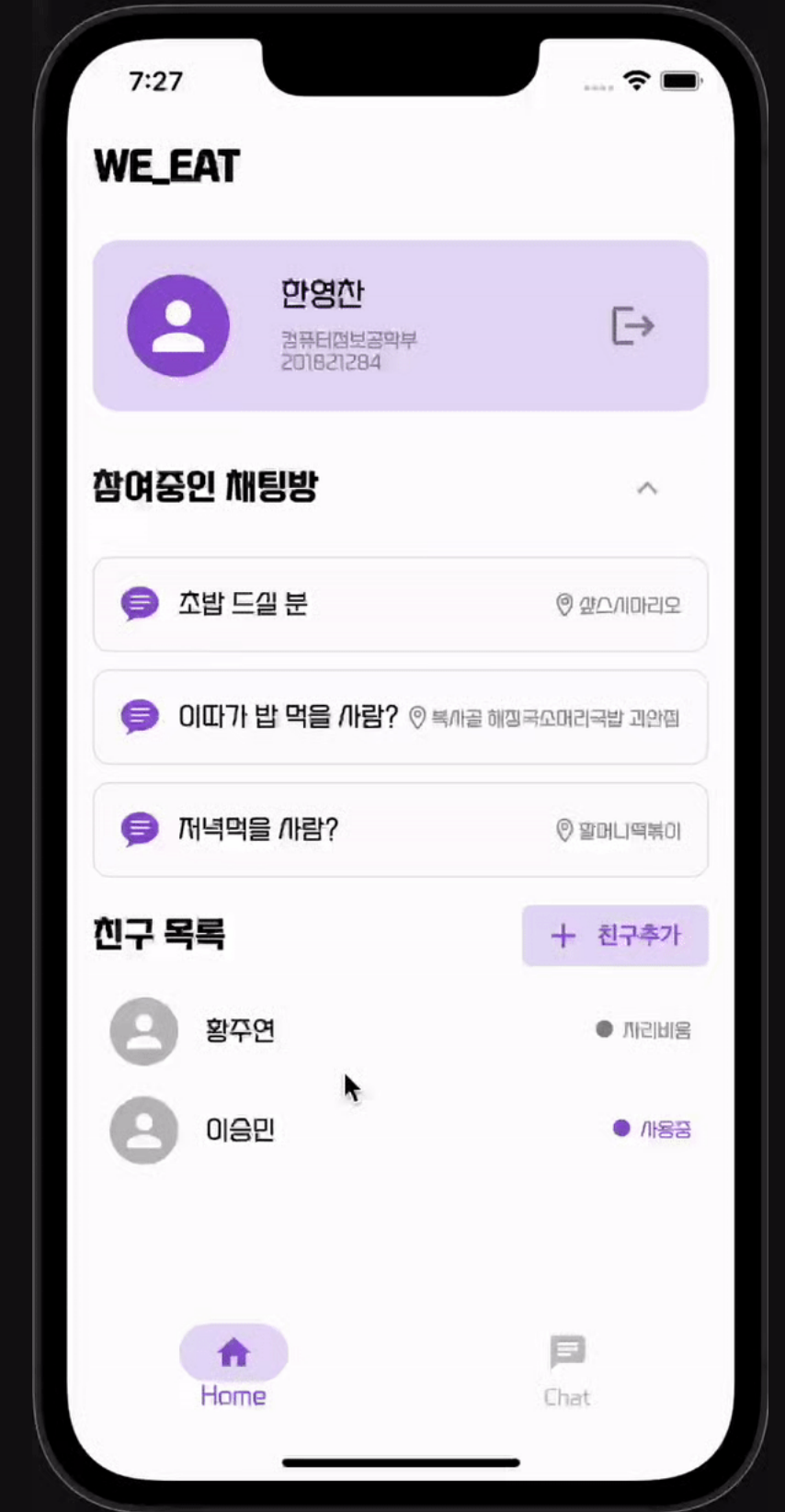
1:n 채팅

사전 조건 : User가 채팅방에 들어와 있는 상태



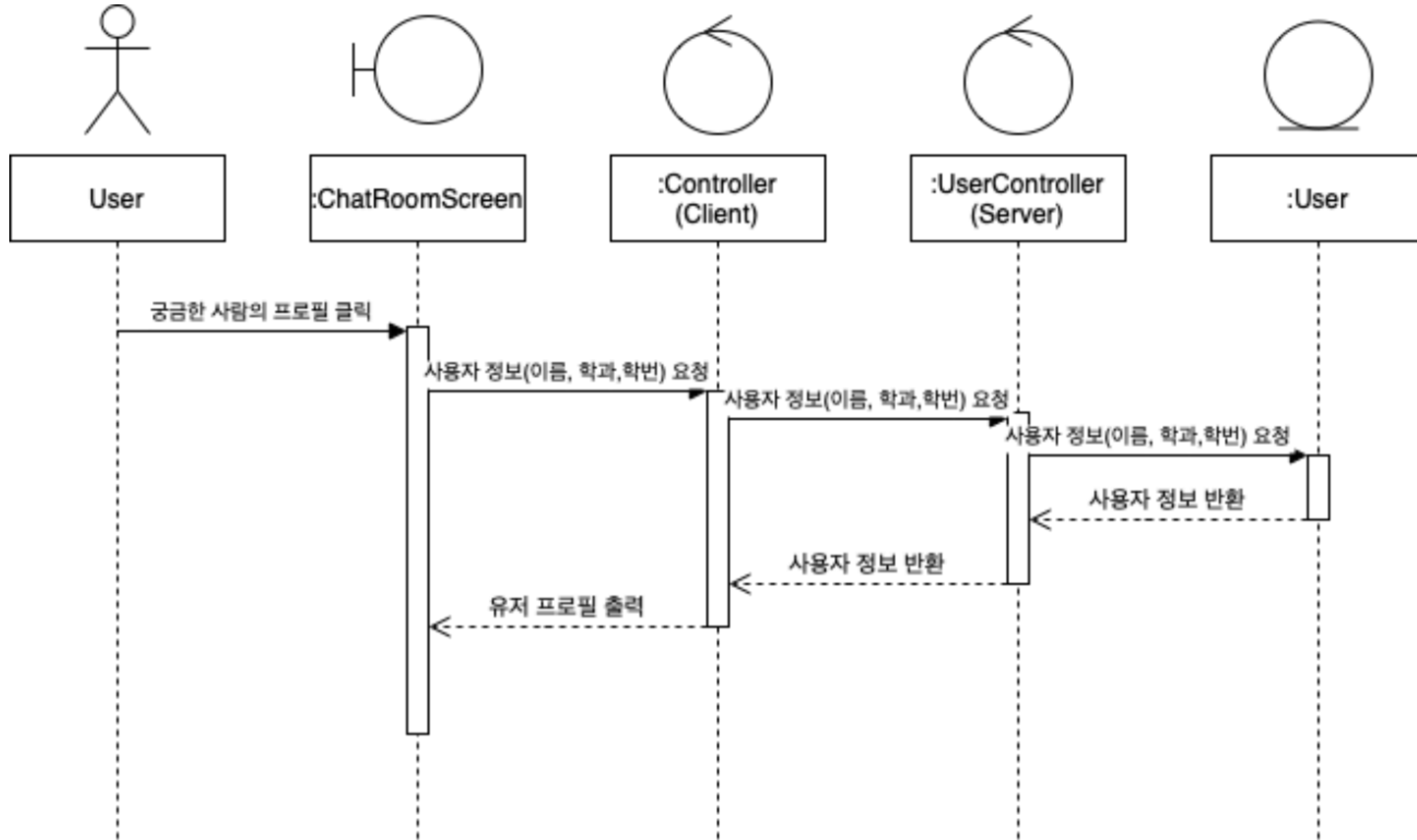
프로필 확인

사용자의 프로필을 클릭하여 정보를 확인합니다.



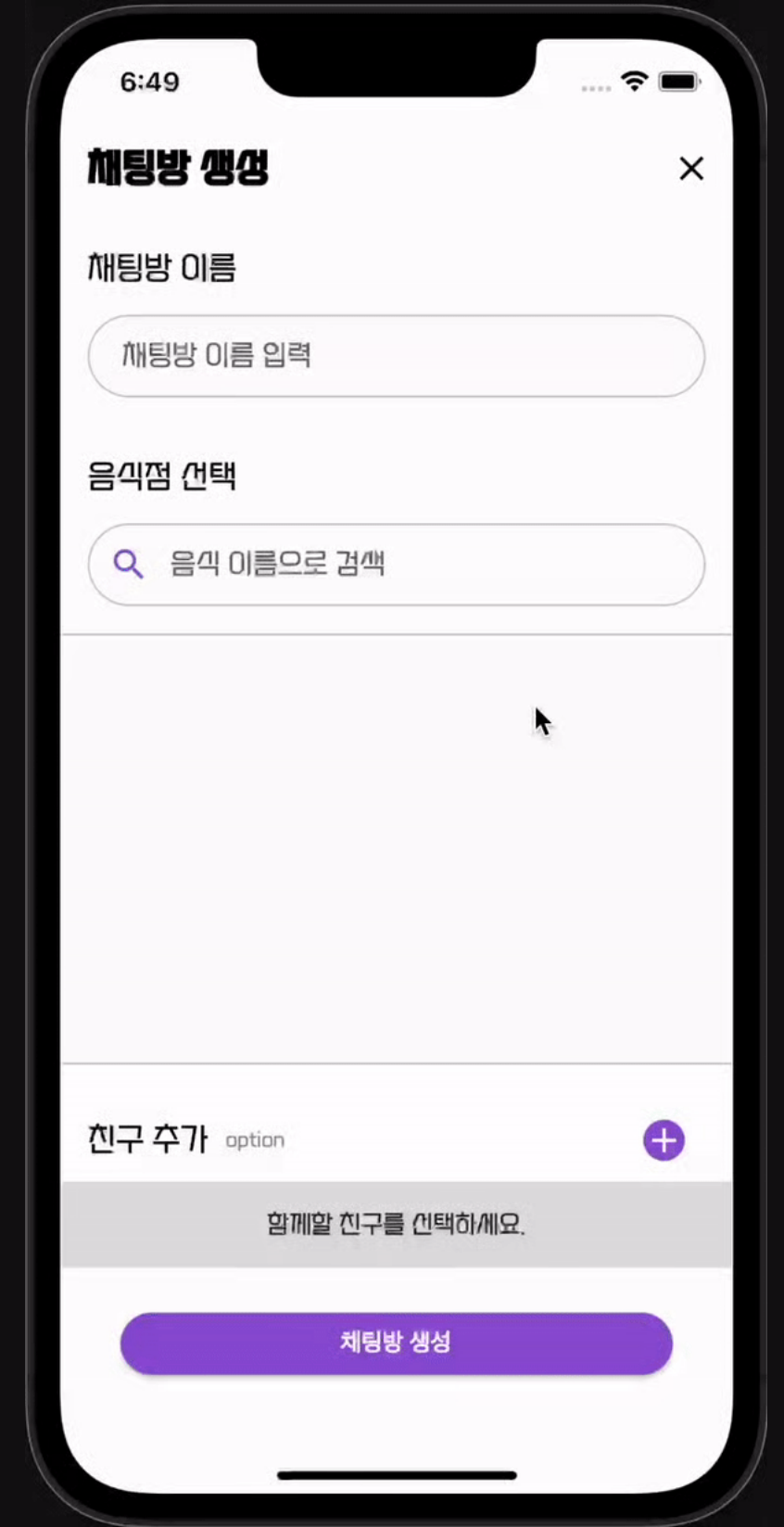
03 Sequence diagram 01. 다이어그램

프로필 확인



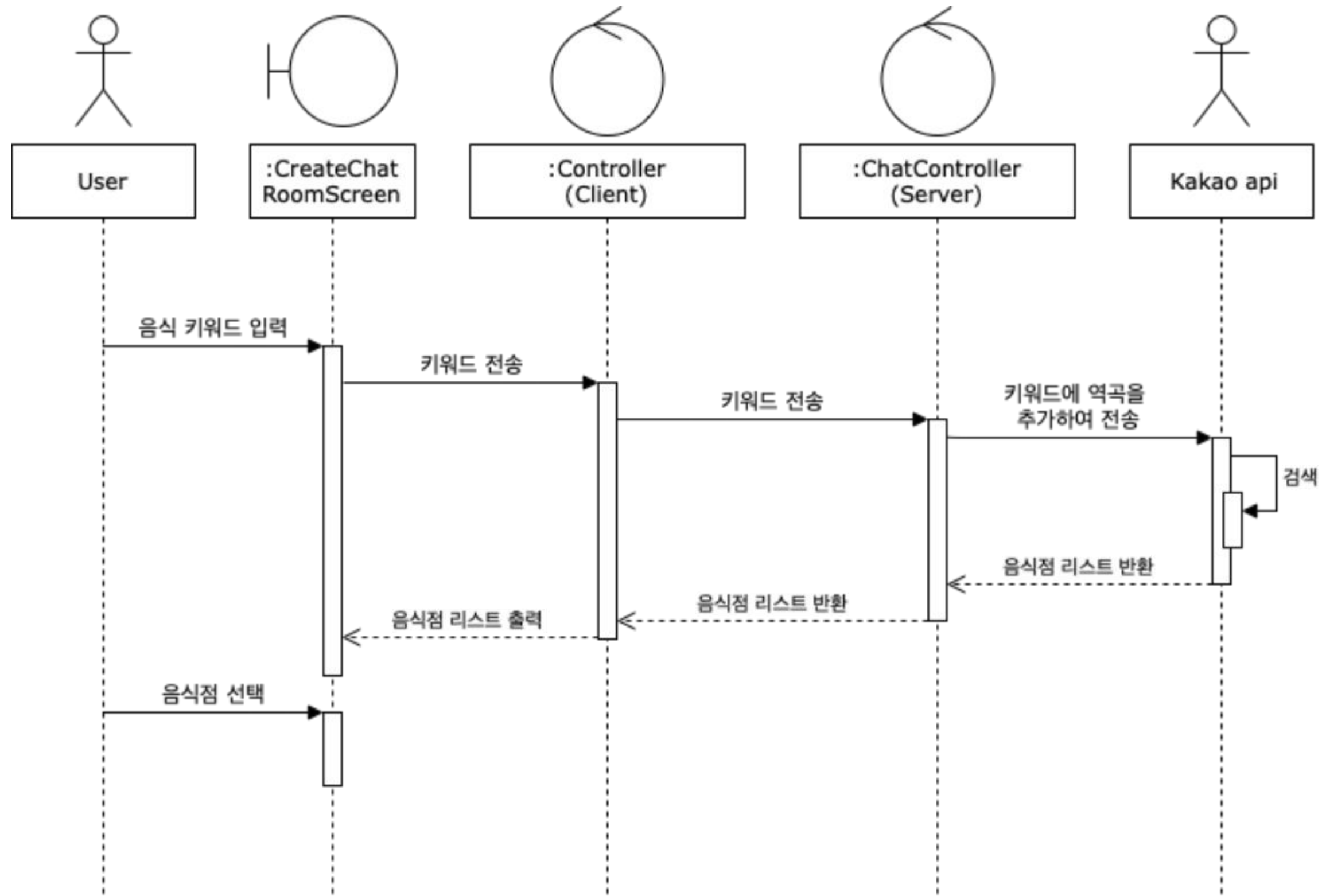
음식점 검색

카카오 API를 이용하여 사용자가 입력한 음식과 관련된 음식점을 검색합니다.



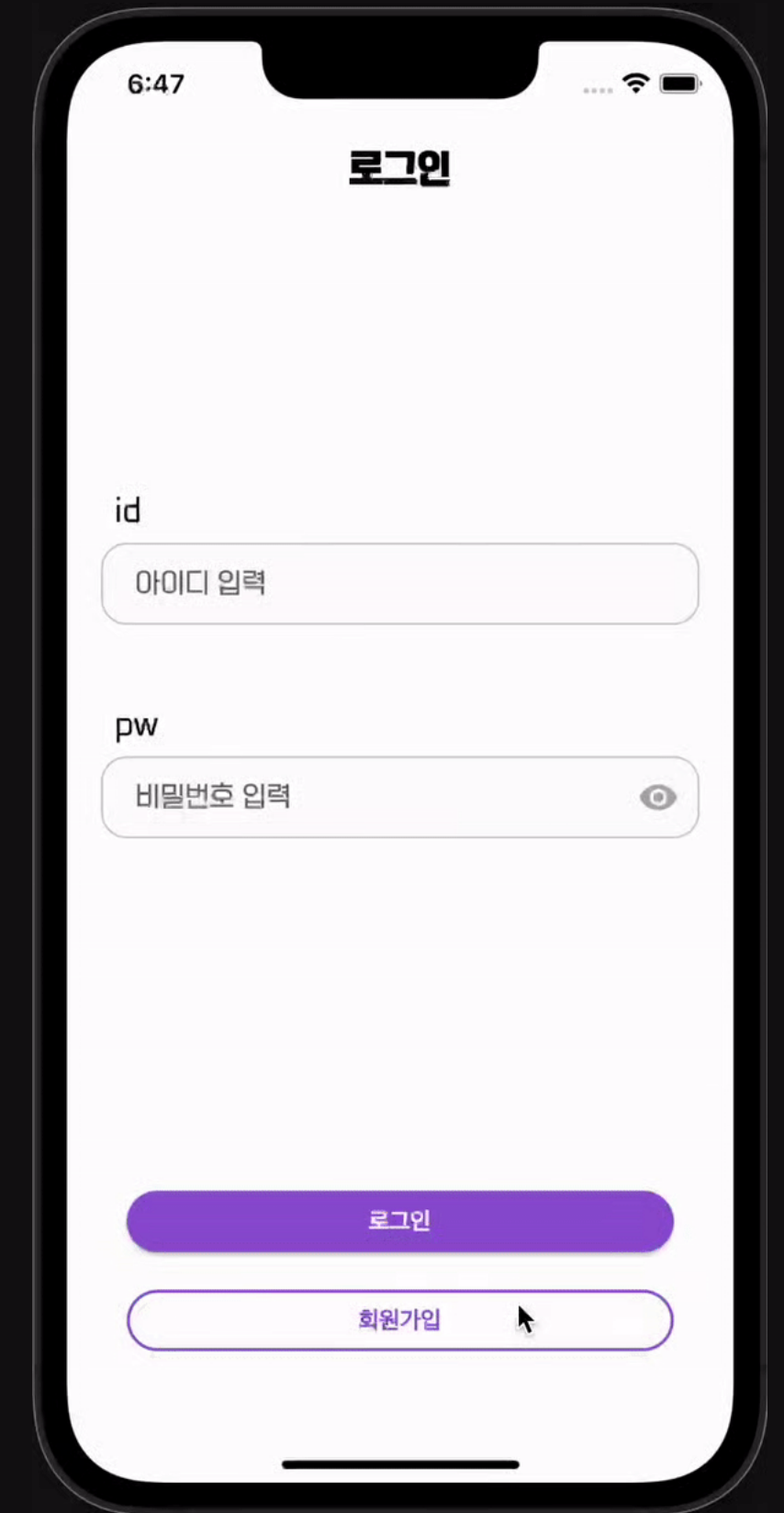
03 Sequence diagram 01. 다이어그램

음식점 검색



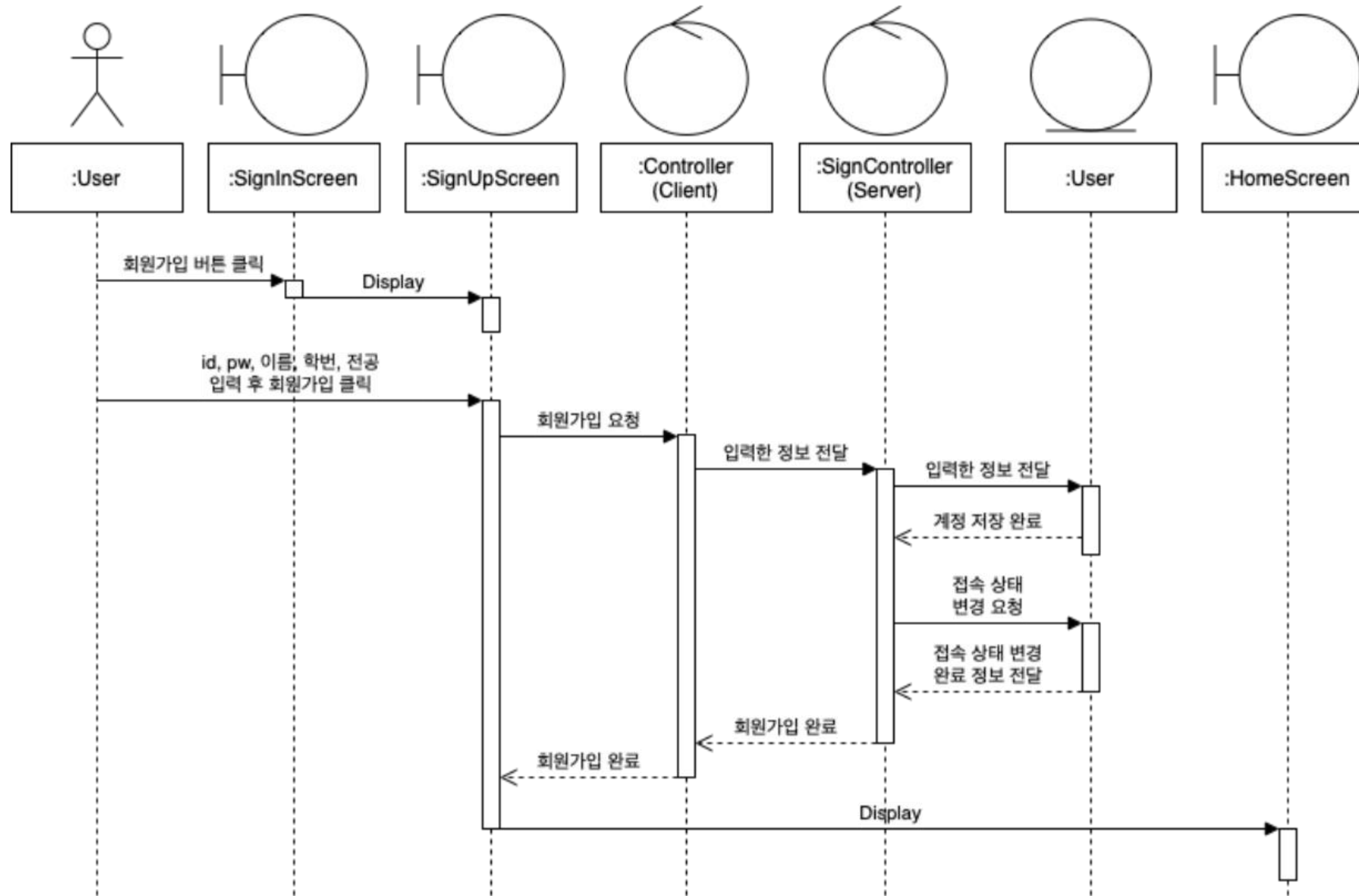
회원가입

사용자 정보(ex. 아이디, 비밀번호, 이름, 학과, 학번)를 입력하고 등록합니다.



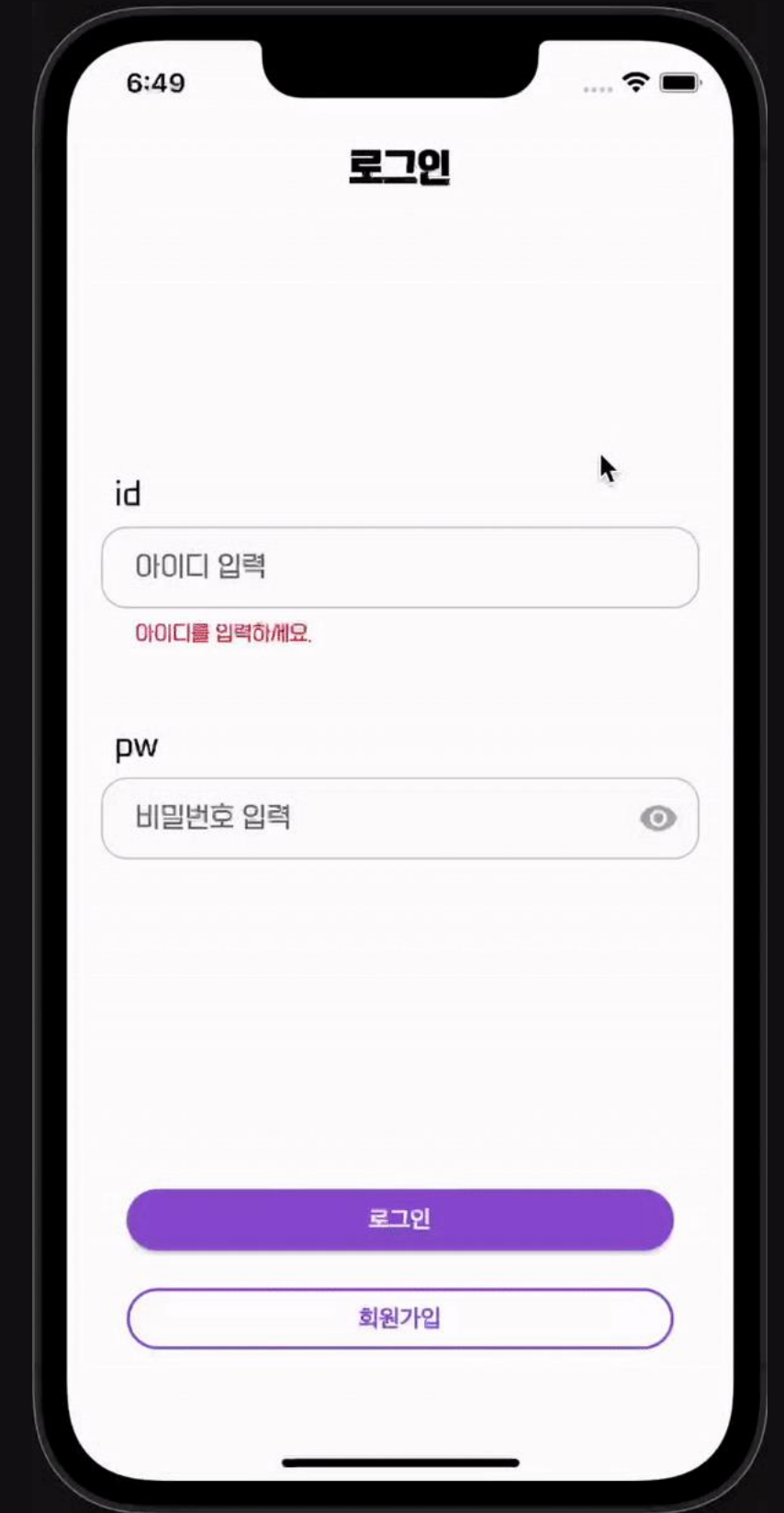
03 Sequence diagram 01. 다이어그램

회원가입



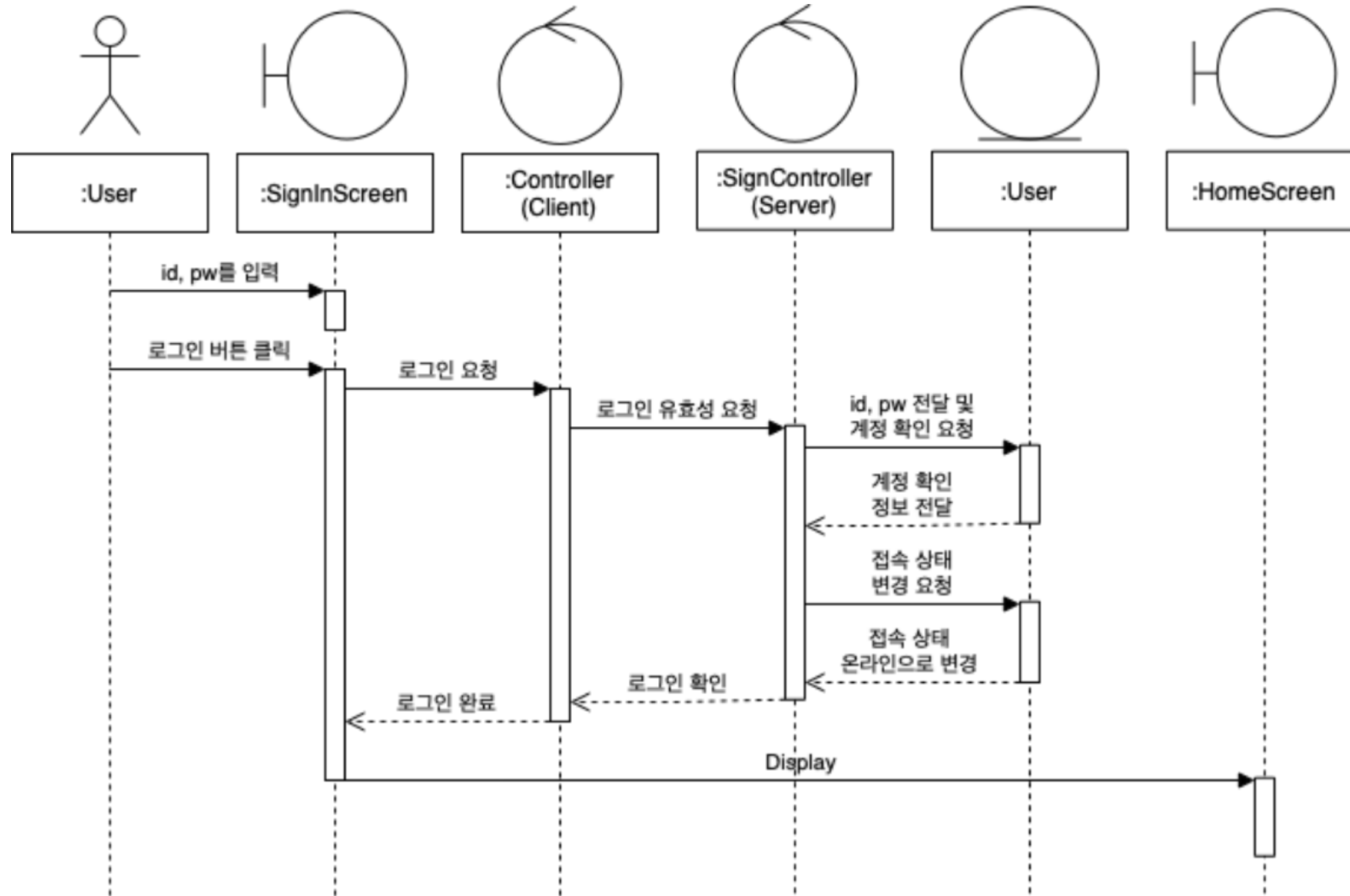
로그인

회원가입한 아이디와 비밀번호를 입력해서
사용자를 온라인 상태로 변경합니다.



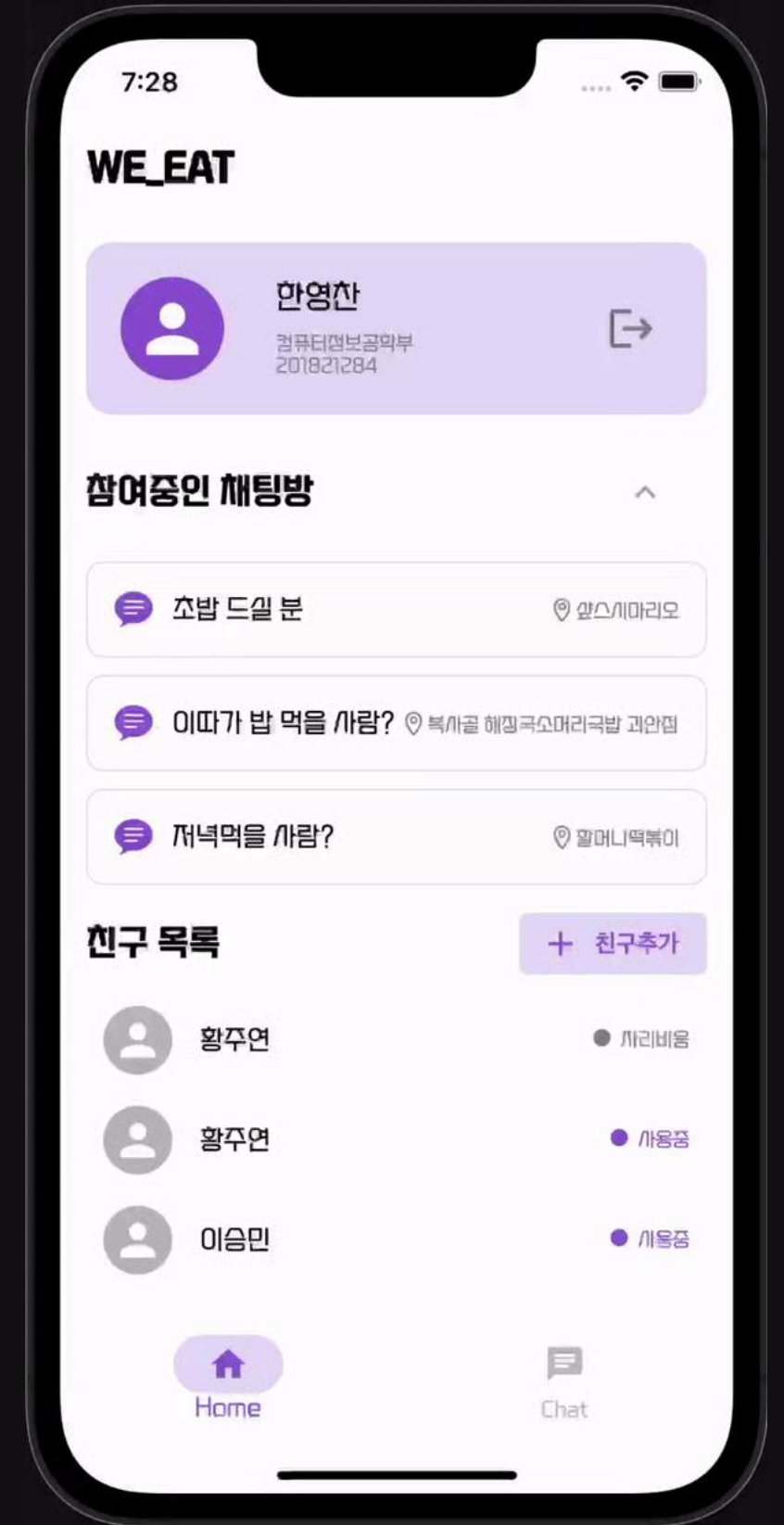
03 Sequence diagram 01. 다이어그램

로그인



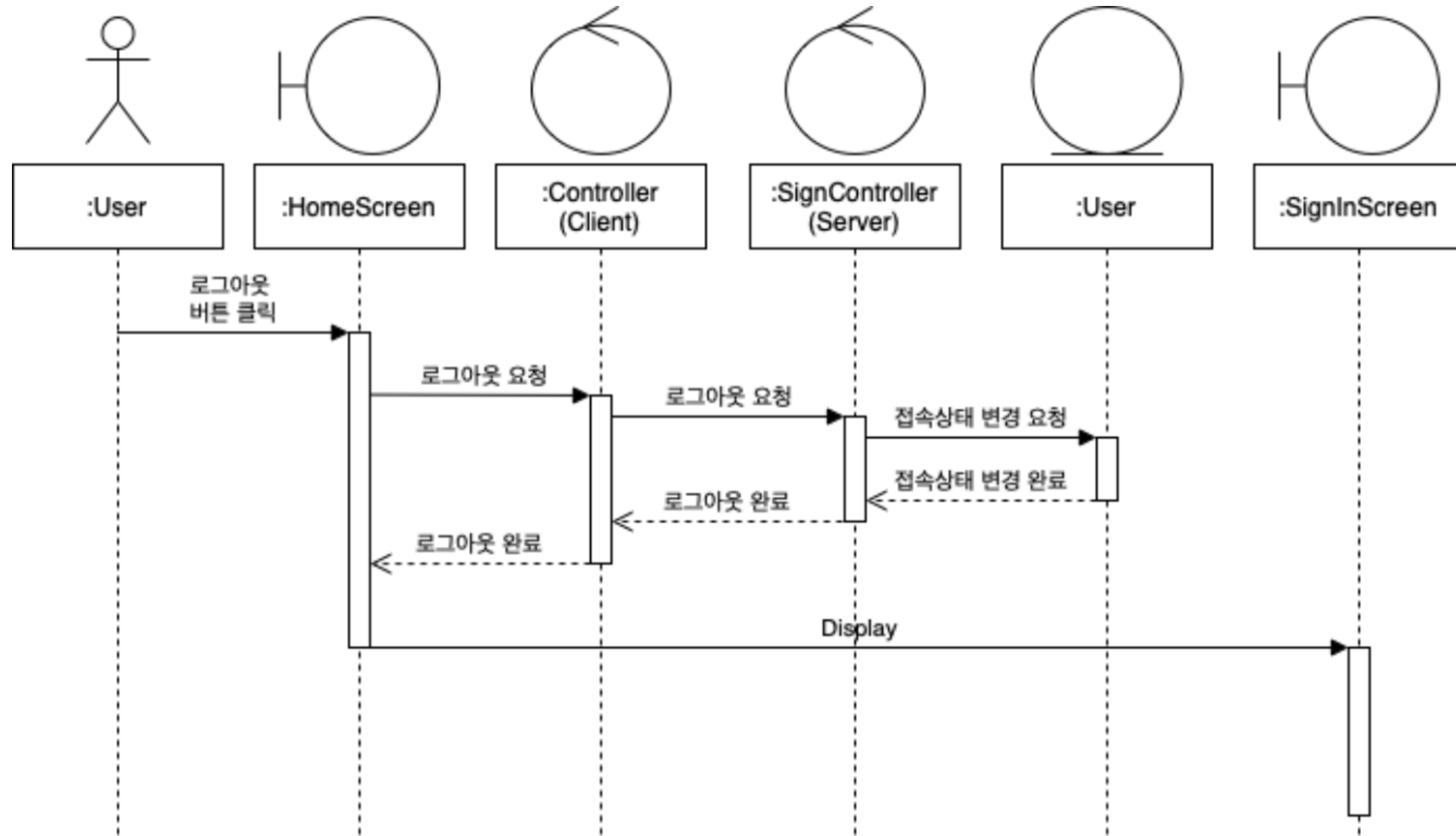
로그아웃

홈 화면의 로그아웃 버튼을 눌러 사용자를 오프라인 상태로 변경합니다.



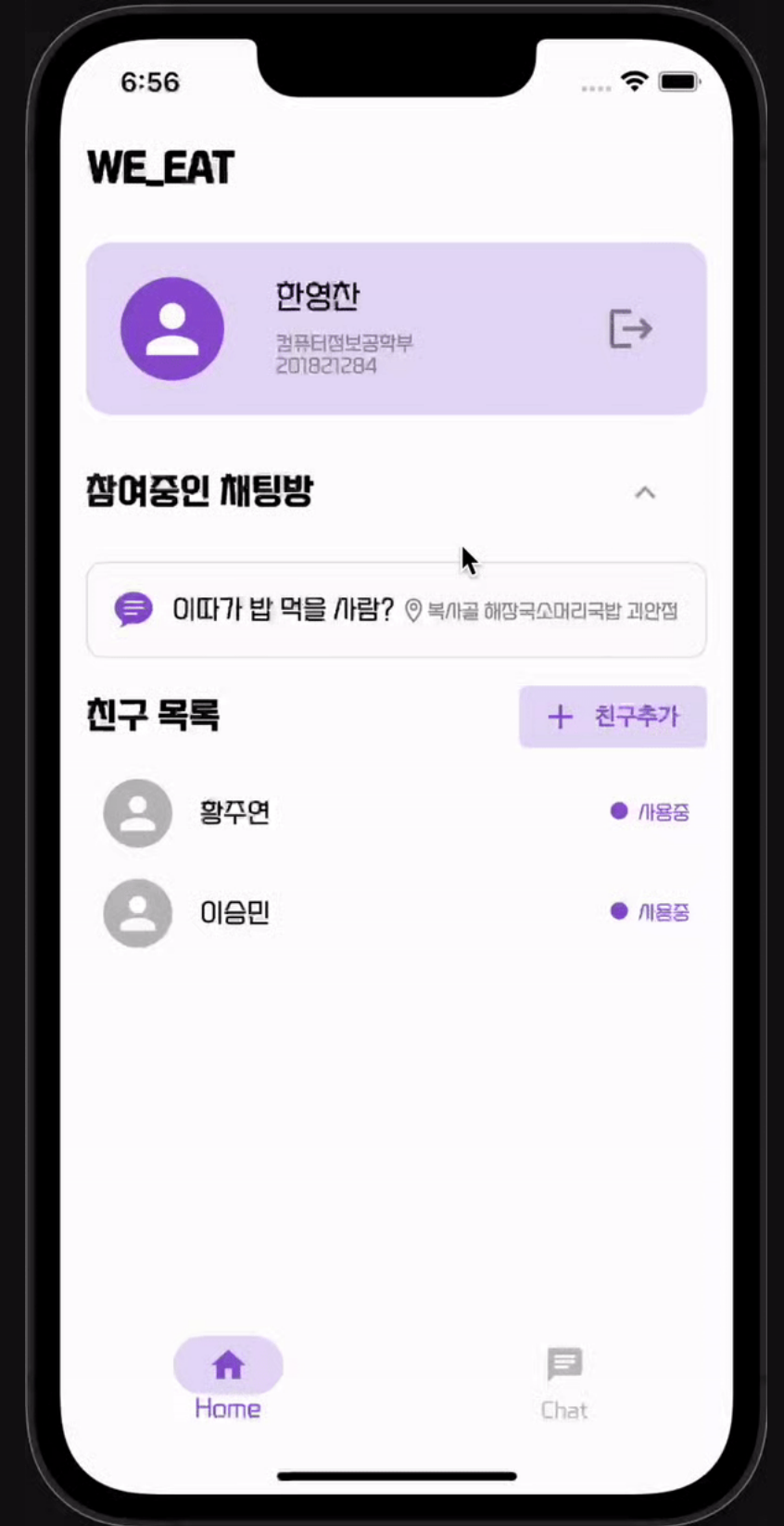
03 Sequence diagram 01. 다이어그램

로그아웃



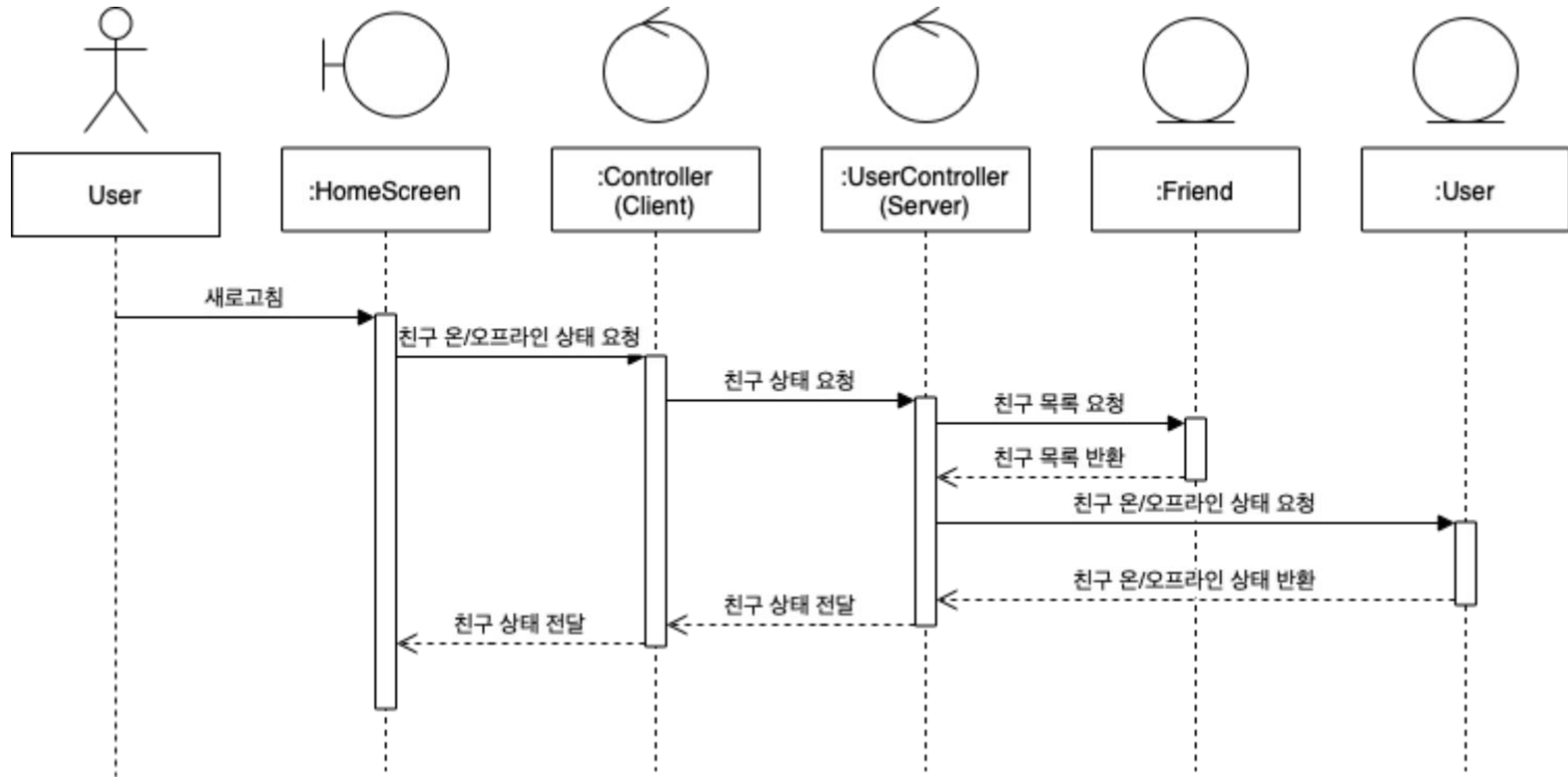
친구 상태 실시간 확인

친구의 현재 온/오프라인 상태를 확인합니다.



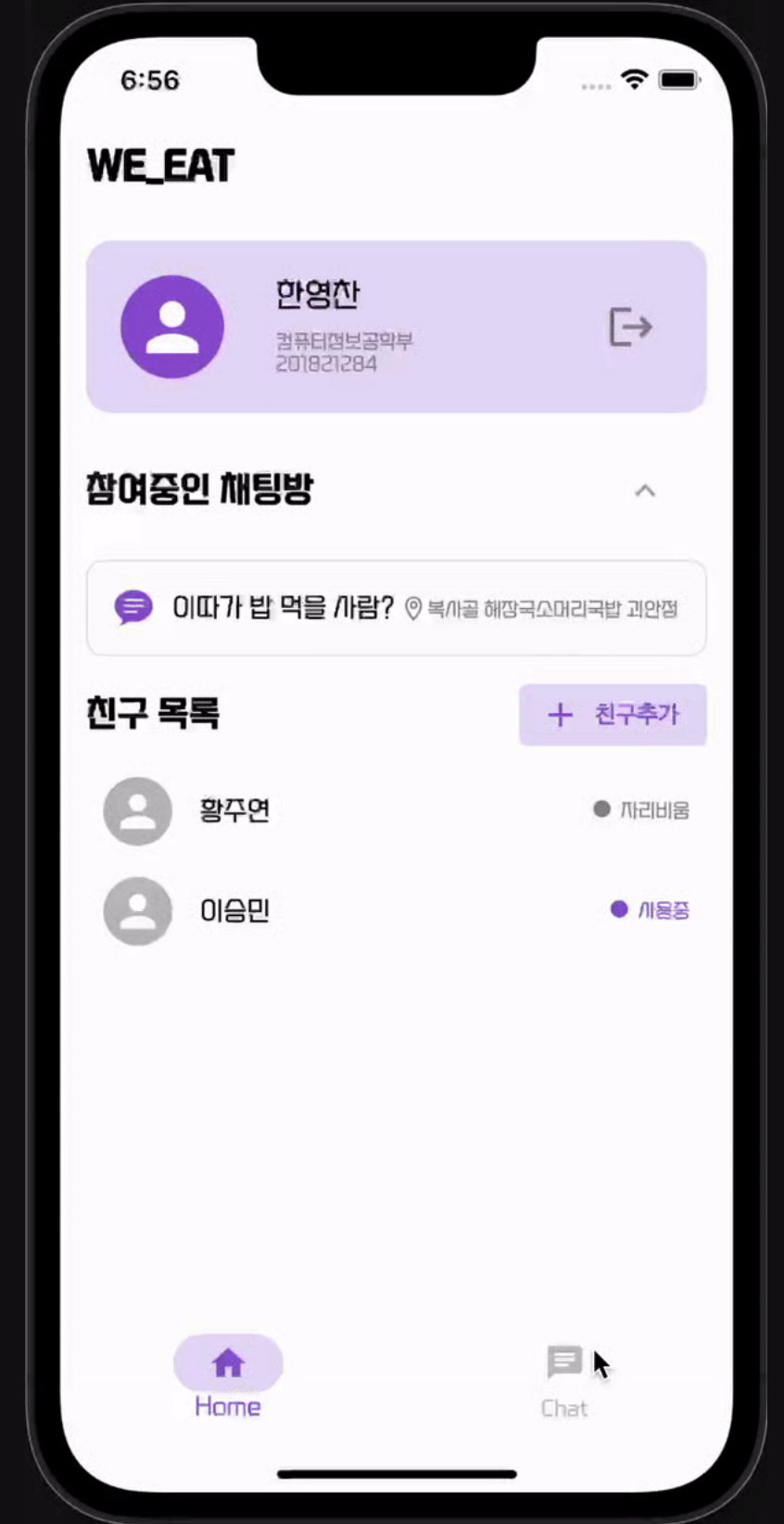
03 Sequence diagram 01. 다이어그램

친구상태 실시간 확인



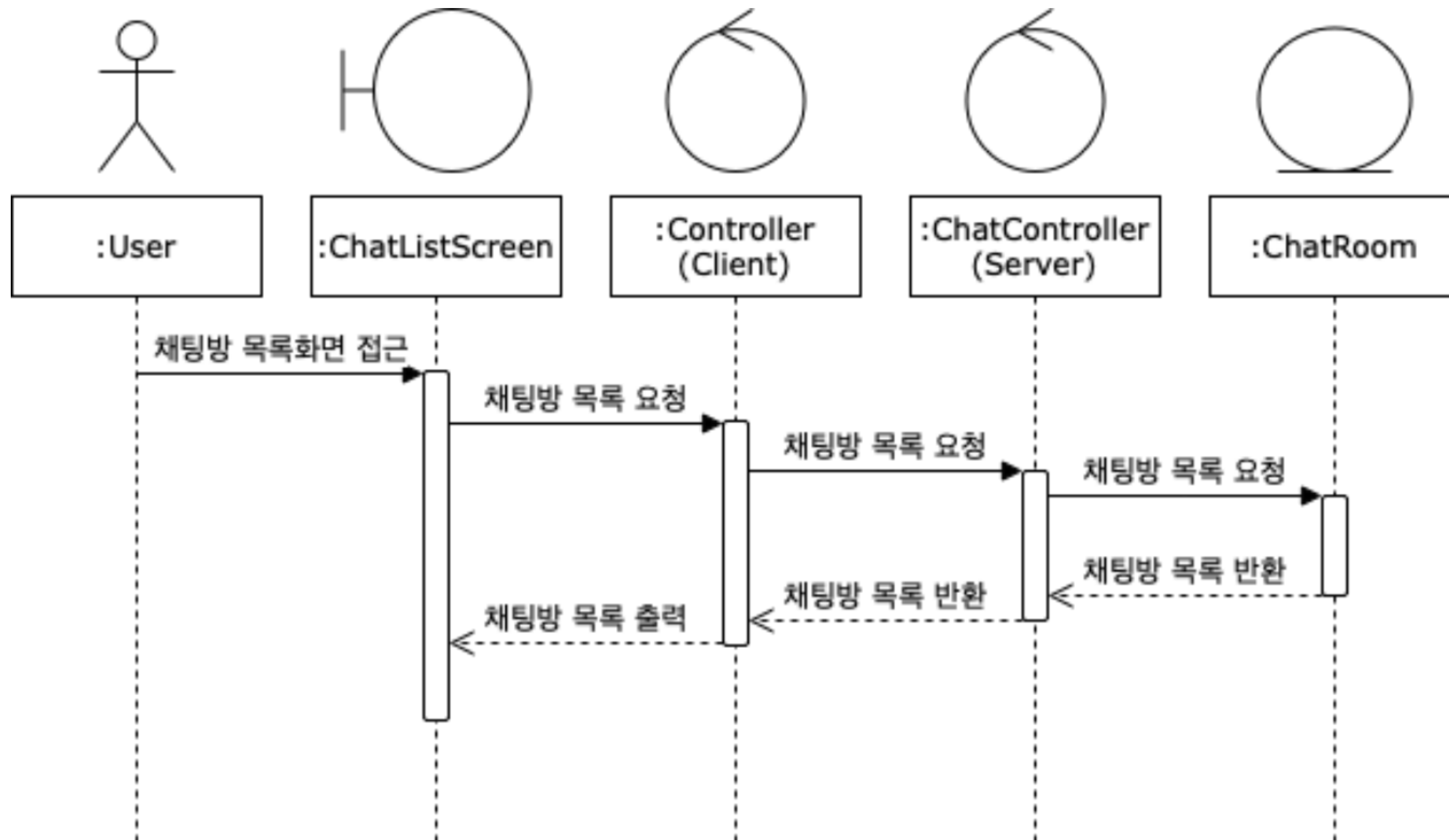
채팅방 목록 확인

현재 생성된 채팅방 목록을 확인합니다.



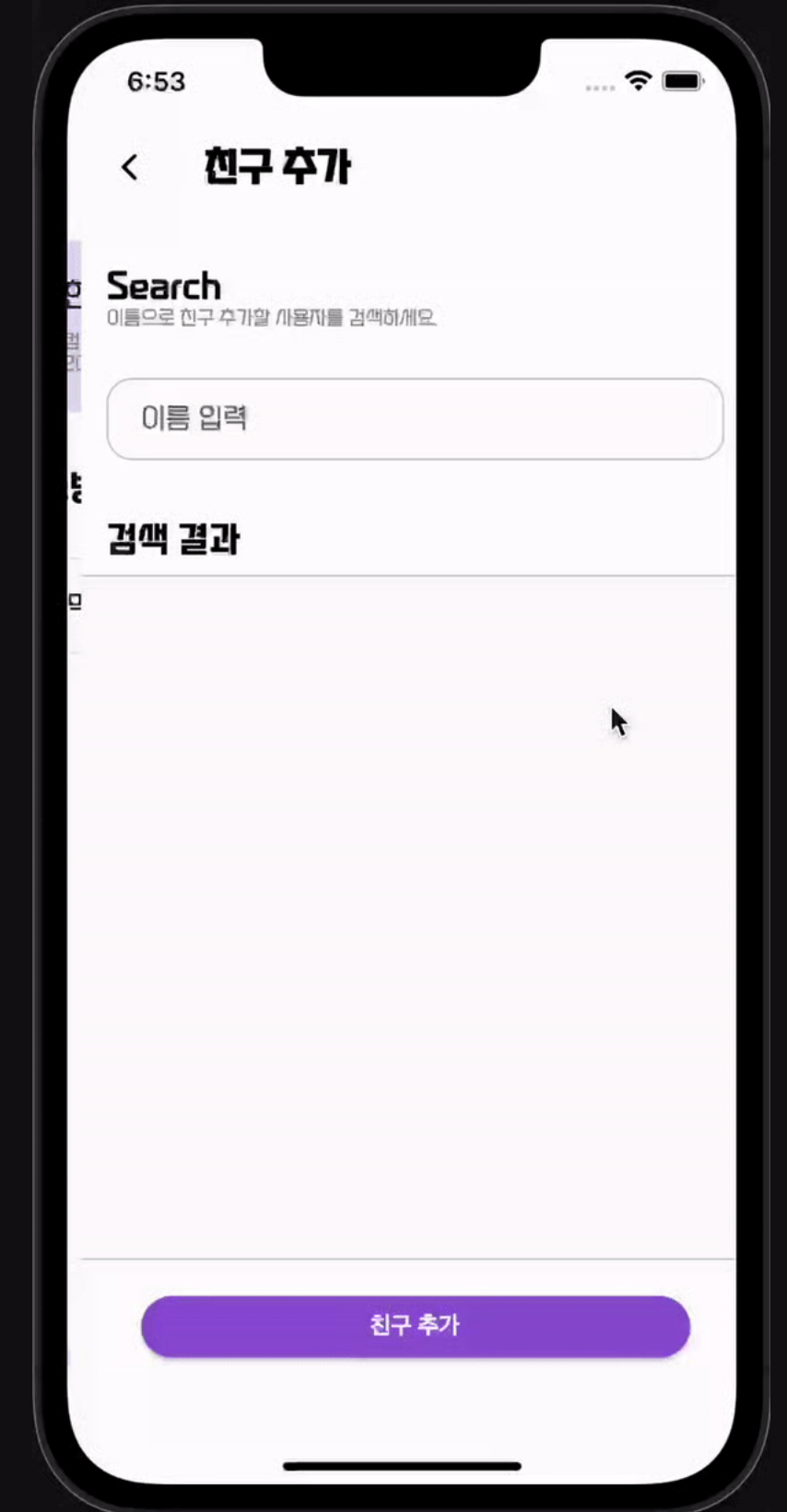
03 Sequence diagram 01. 다이어그램

채팅방 목록 확인



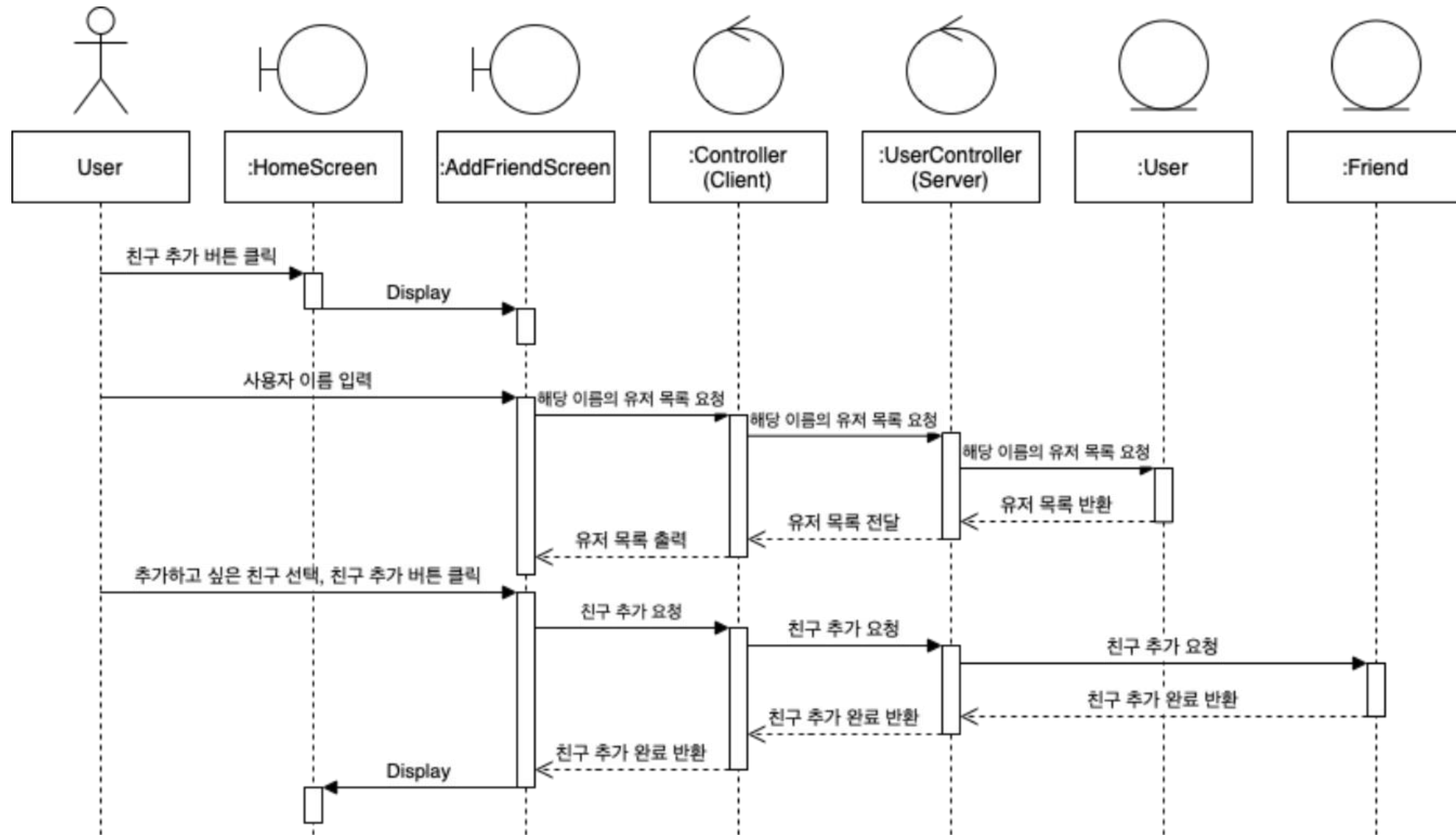
친구 추가

친구의 이름을 검색하여 친구를 추가합니다.



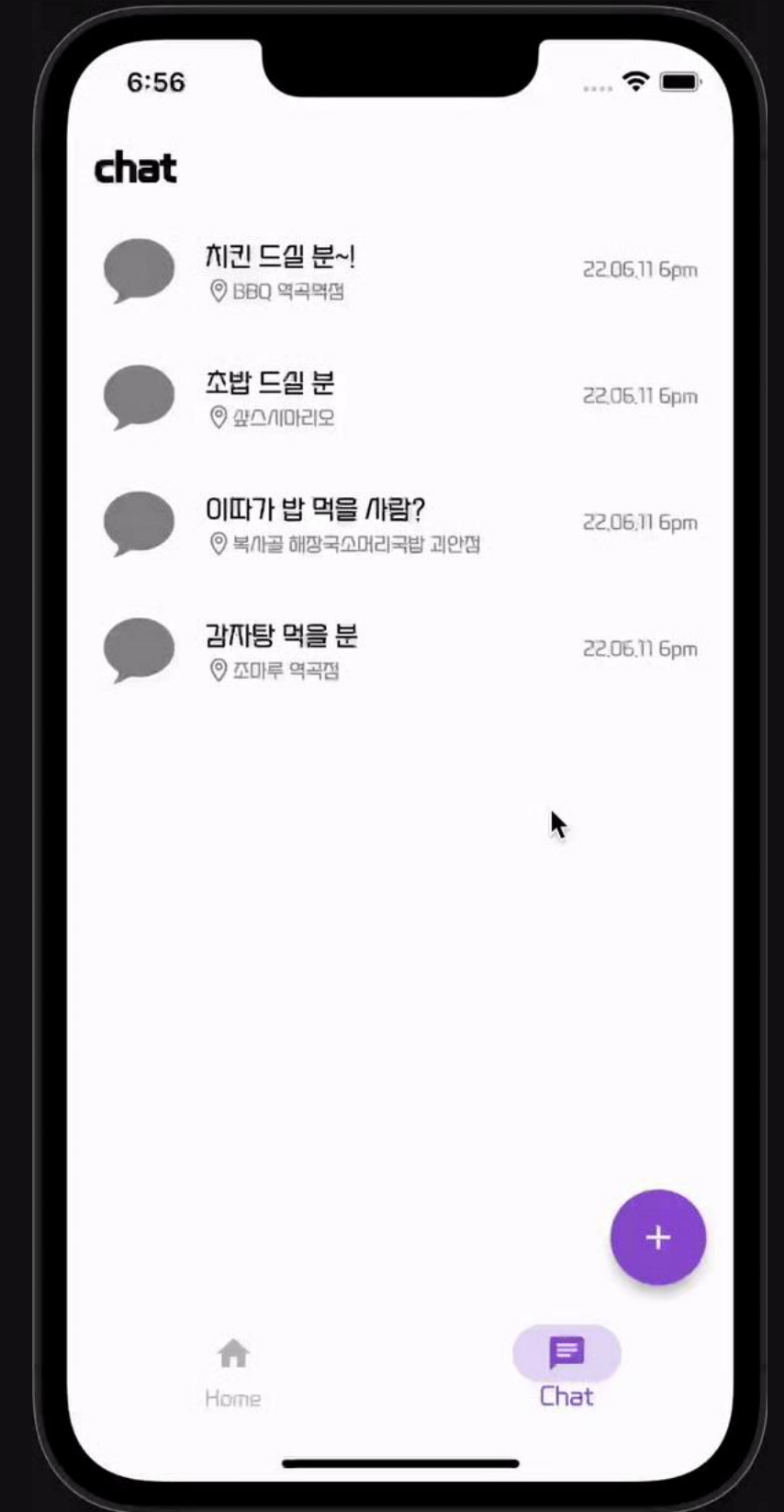
03 Sequence diagram 01. 다이어그램

친구 추가



채팅방 참여1

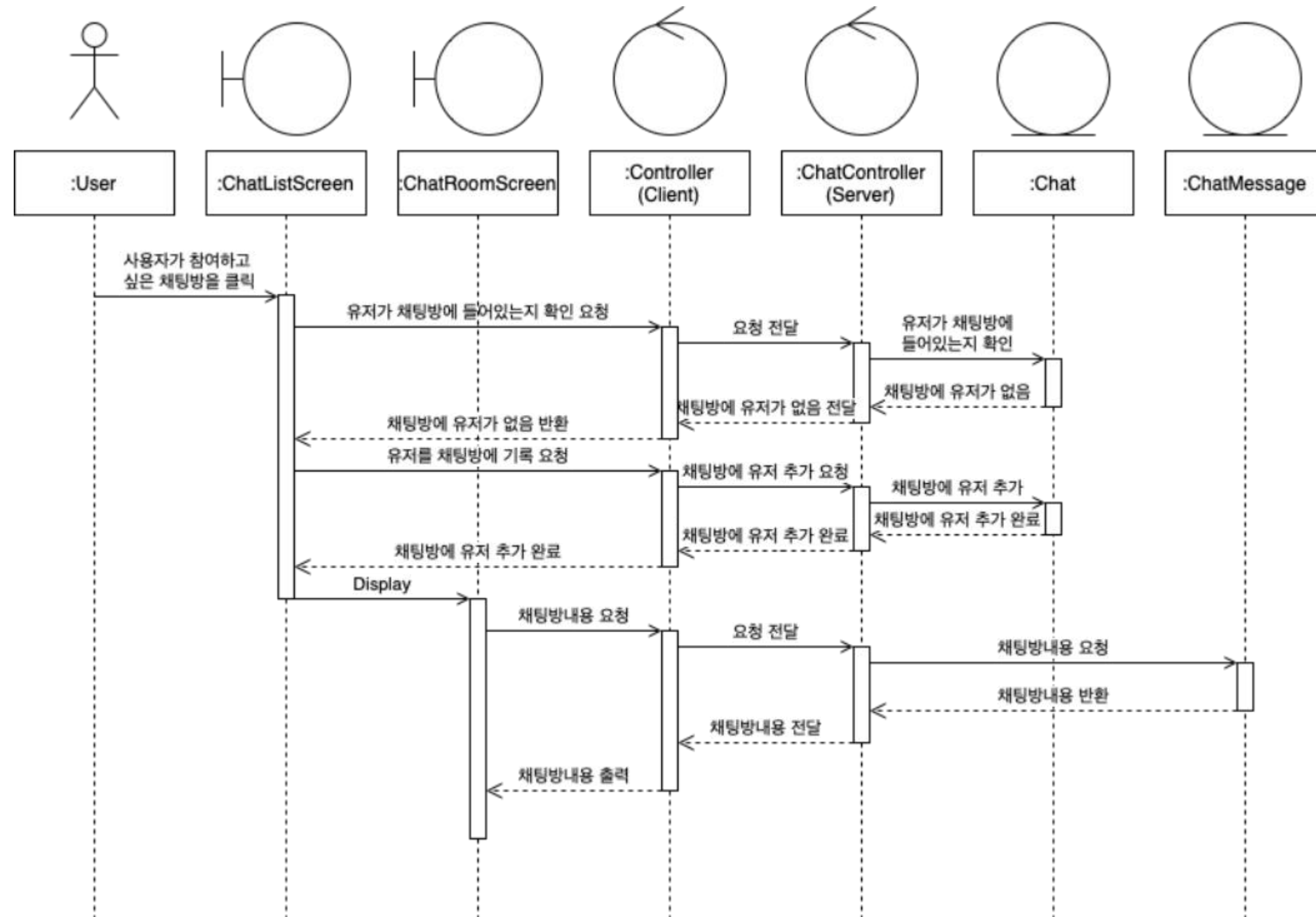
참가하고 싶어하는 채팅방을 클릭하여 참가합니다.
(채팅방에 처음 들어가는 경우)



03 Sequence diagram 01. 다이어그램

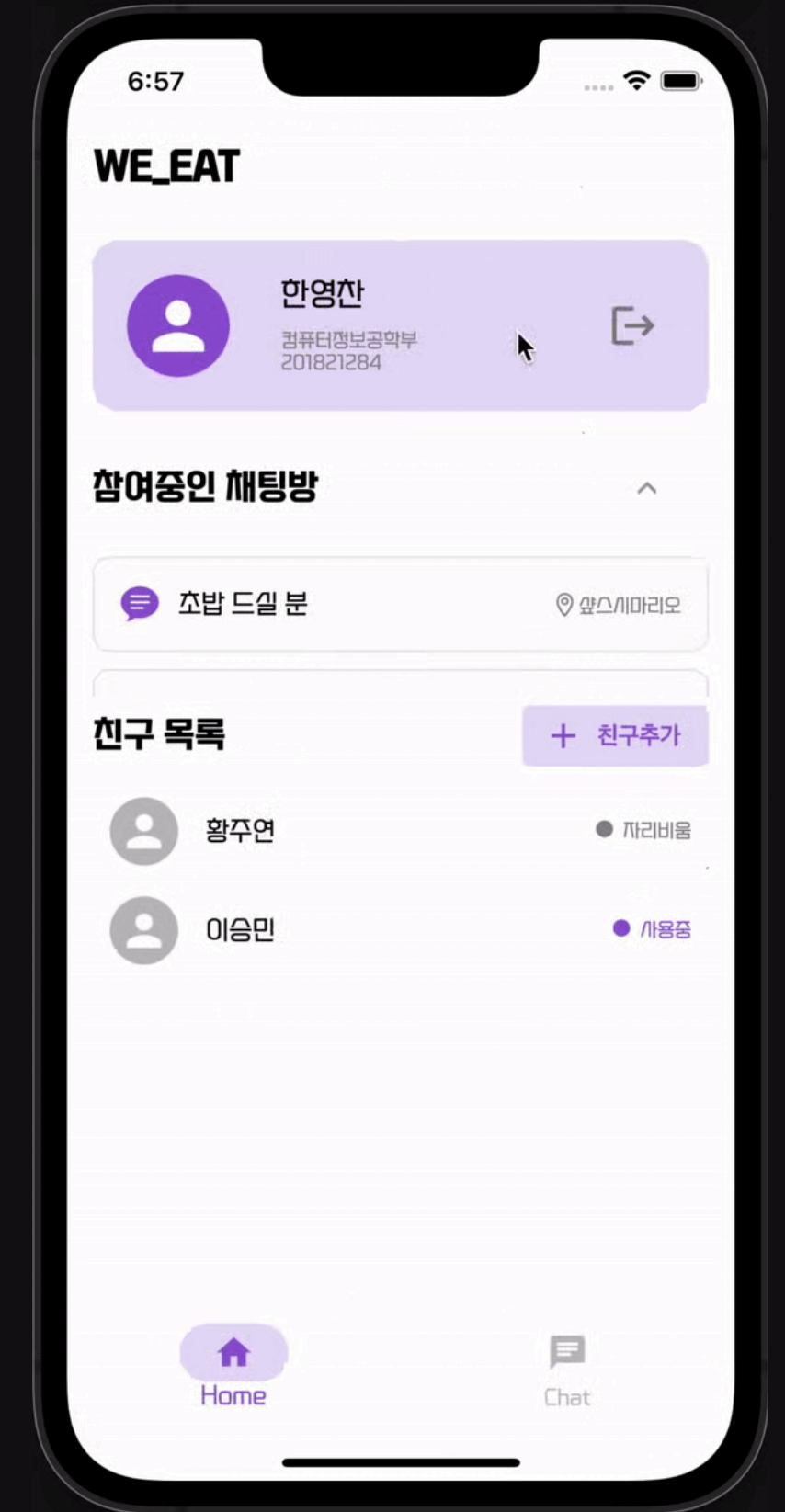
채팅방 참여1

채팅방에 처음 들어가는 경우



채팅방 참여2

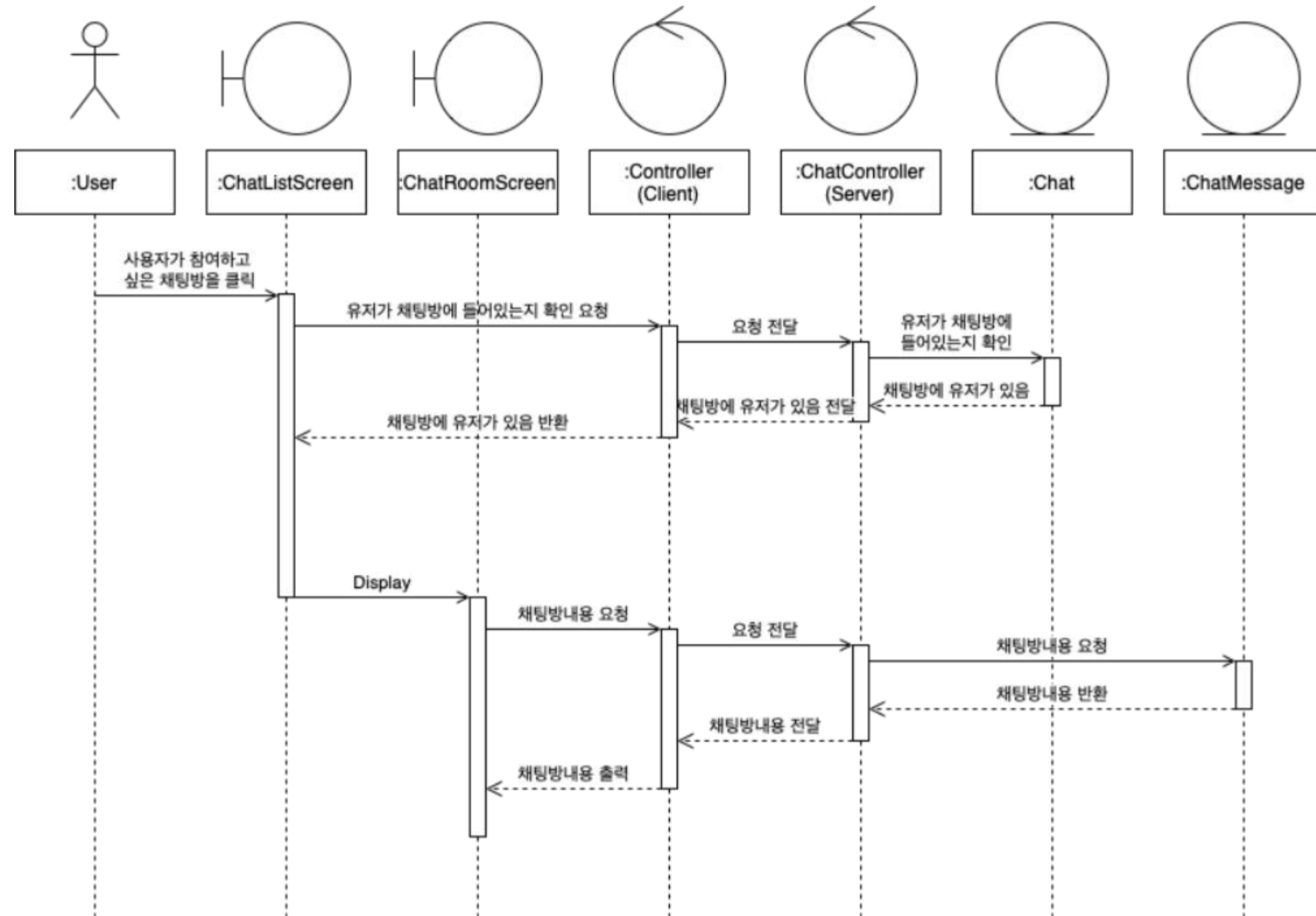
참가하고 싶어하는 채팅방을 클릭하여 참가합니다.
(채팅방에 참여하고 있는 경우)



03 Sequence diagram 01. 다이어그램

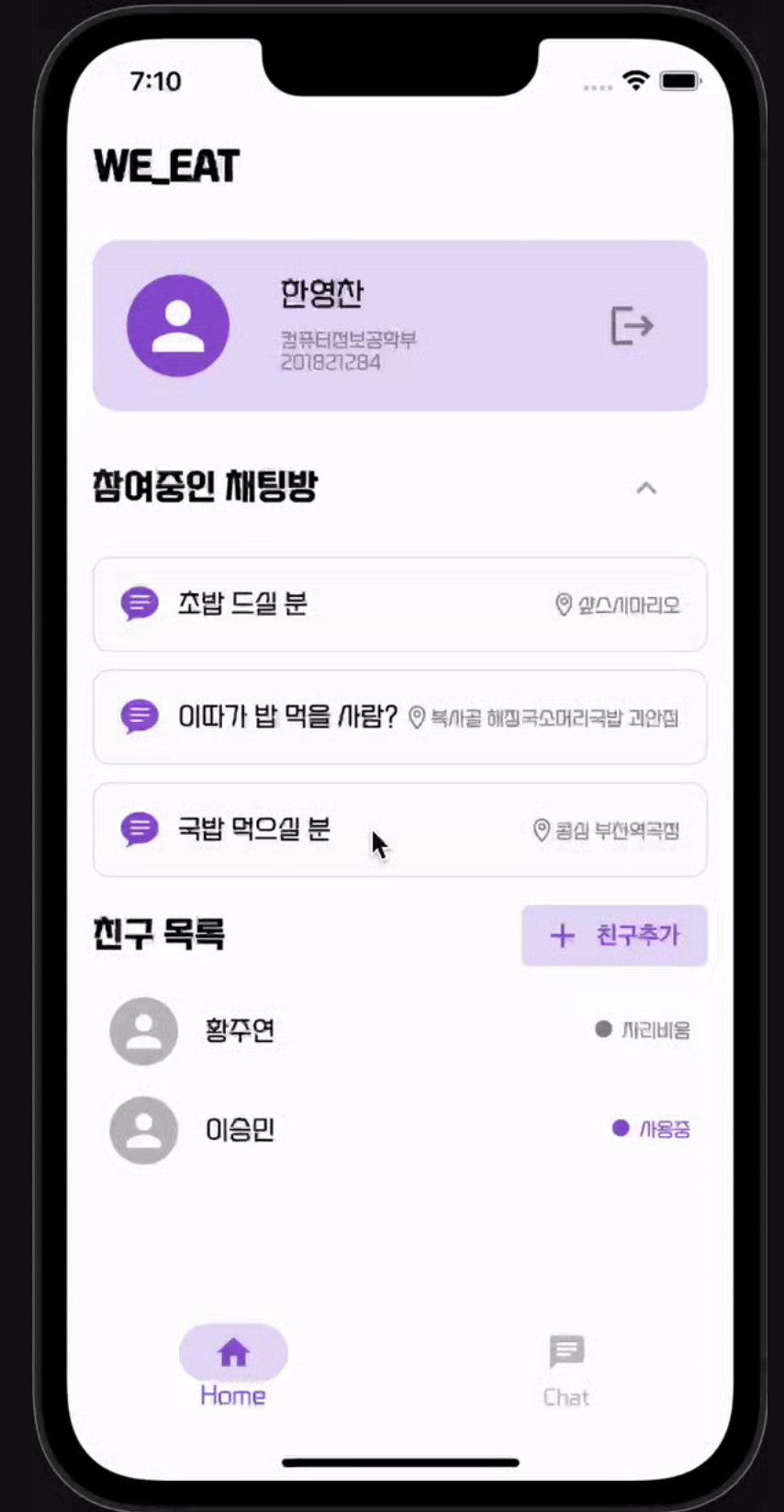
채팅방 참여2

해당 채팅방에 참여하고 있는 경우



채팅방 나가기1

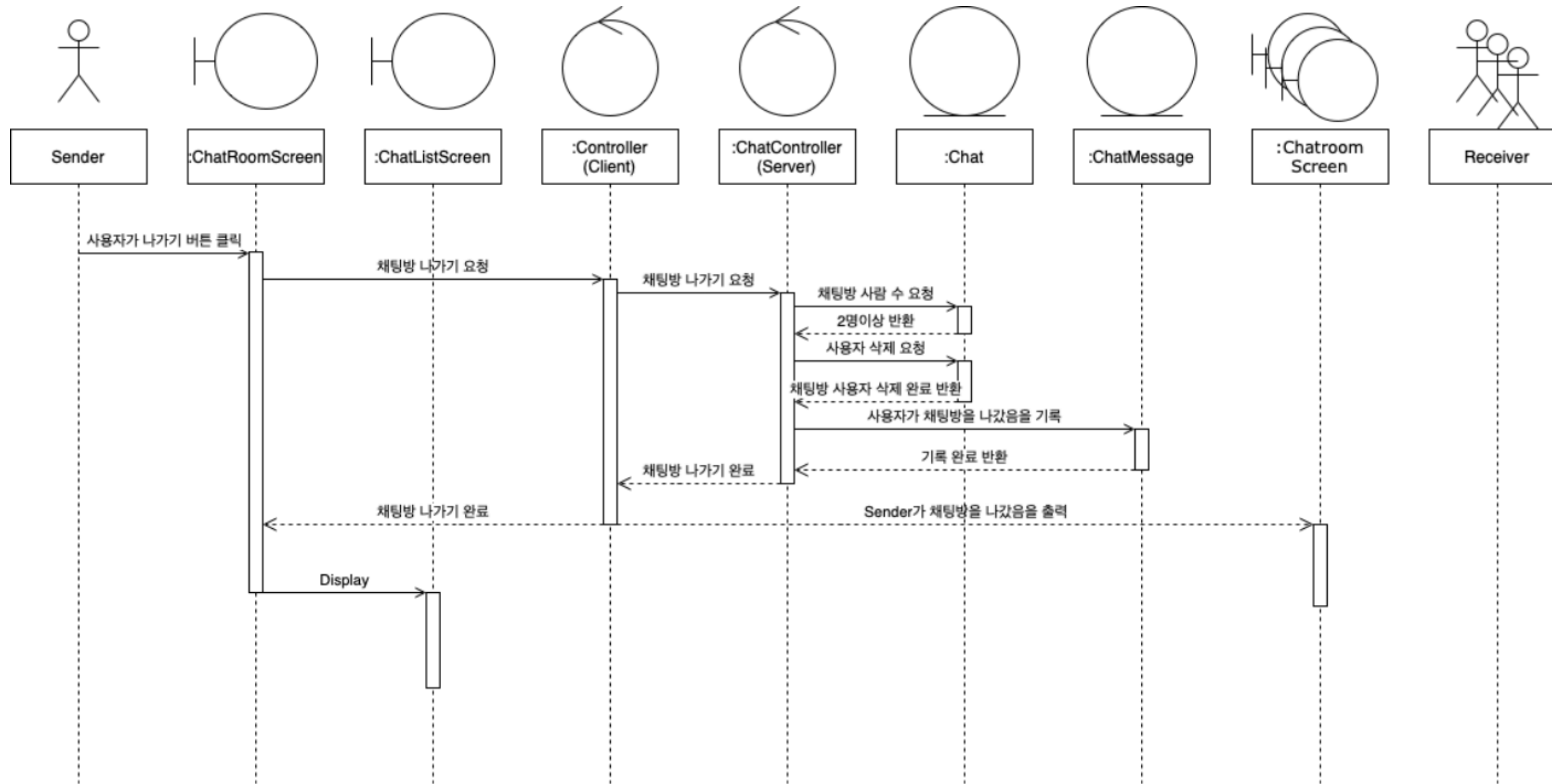
채팅방에서 나가기 버튼을 클릭하여 채팅방을 나갑니다.
(사용자가 채팅방을 나갈 때, 남아 있는 사용자가 있는 경우)



03 Sequence diagram 01. 다이어그램

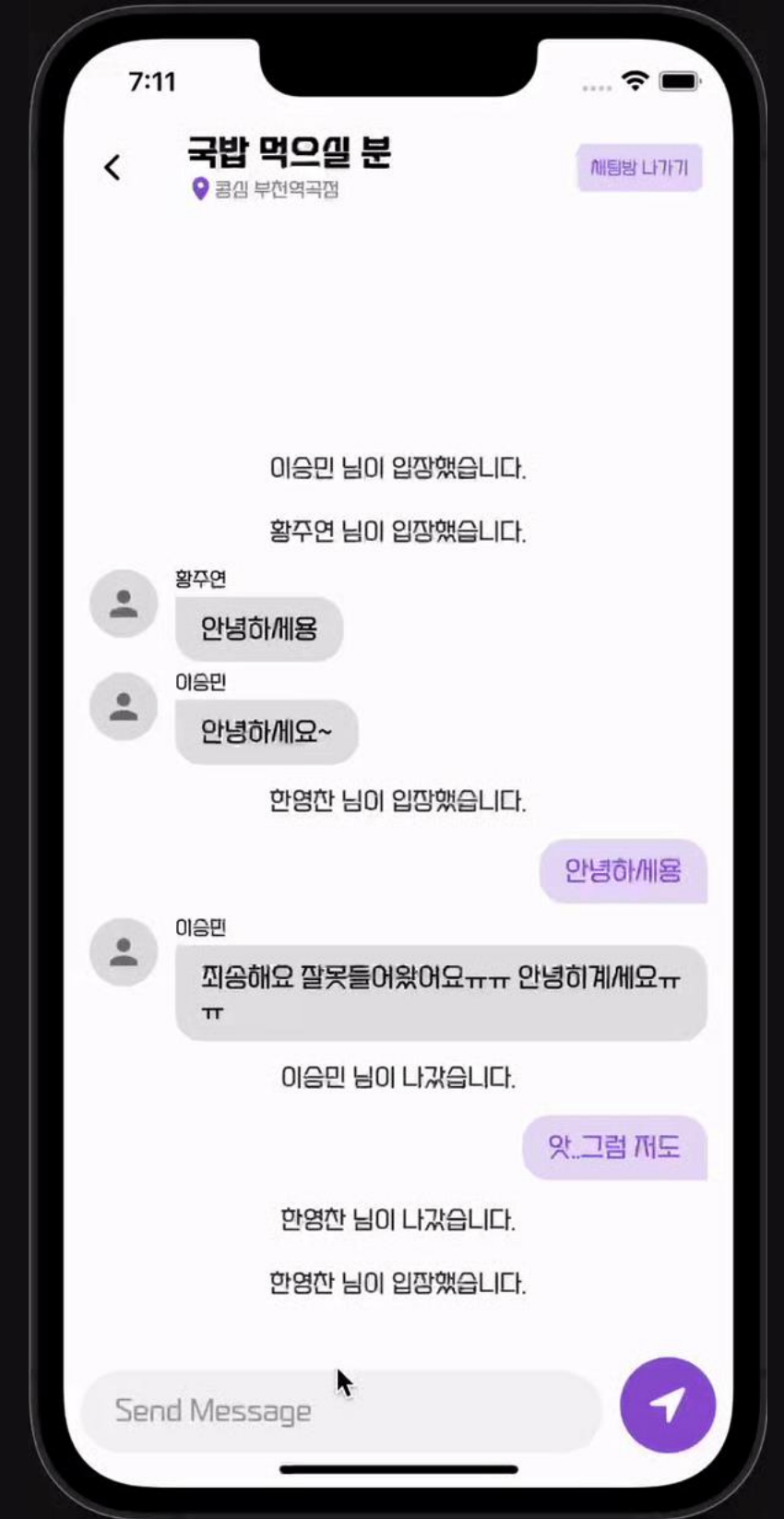
채팅방 나가기1

사용자가 채팅방을 나갈때, 남아 있는 사용자가 있는 경우



채팅방 나가기2

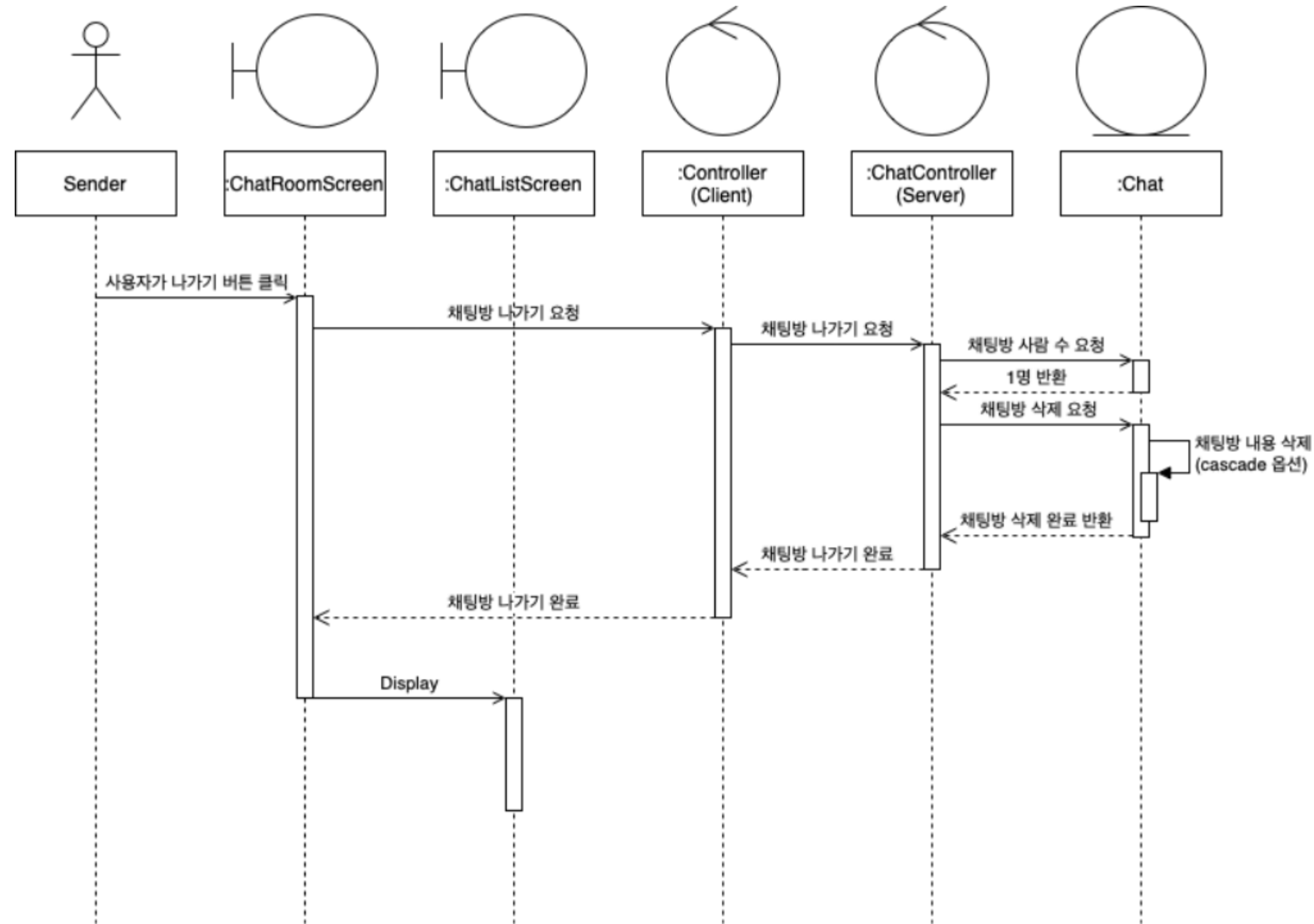
채팅방에서 나가기 버튼을 클릭하여 채팅방을 나갑니다.
(사용자가 채팅방을 나갈 때, 남아 있는 사용자가 없는 경우)



03 Sequence diagram 01. 다이어그램

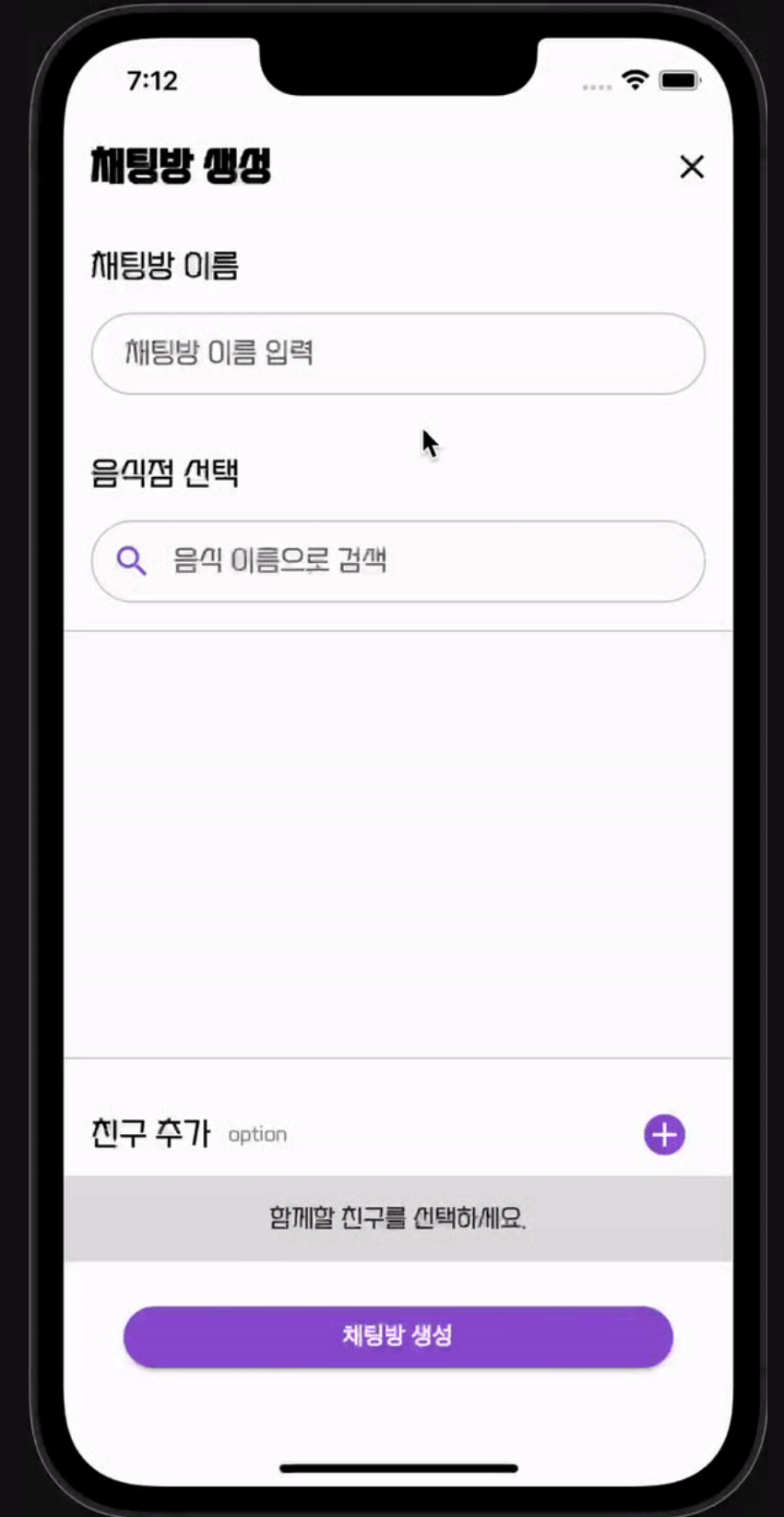
채팅방 나가기2

사용자가 채팅방을 나갈때, 남아 있는 사용자가 없는 경우



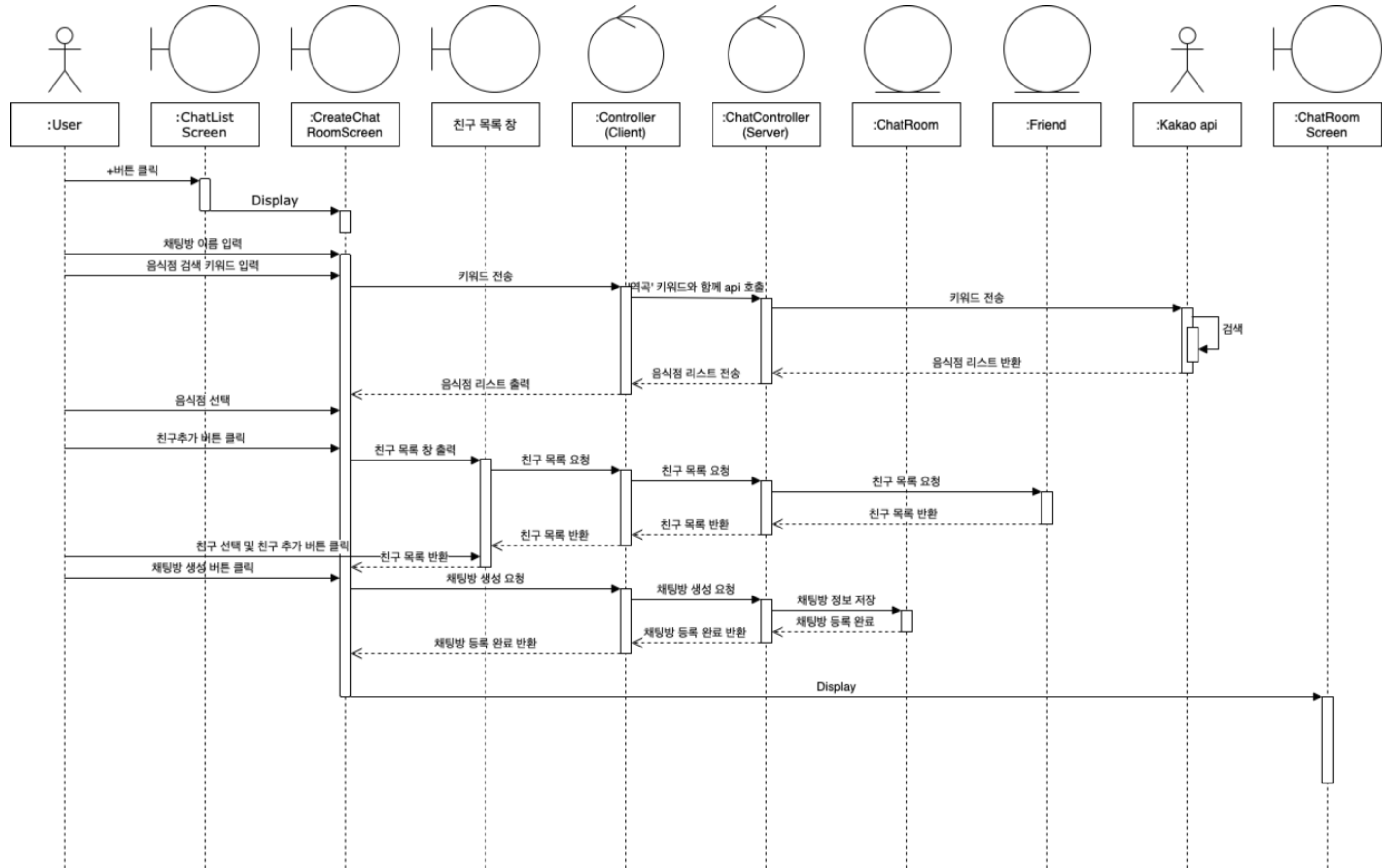
채팅방 생성

역곡 주변 음식점을 기준으로 채팅방을 생성합니다.



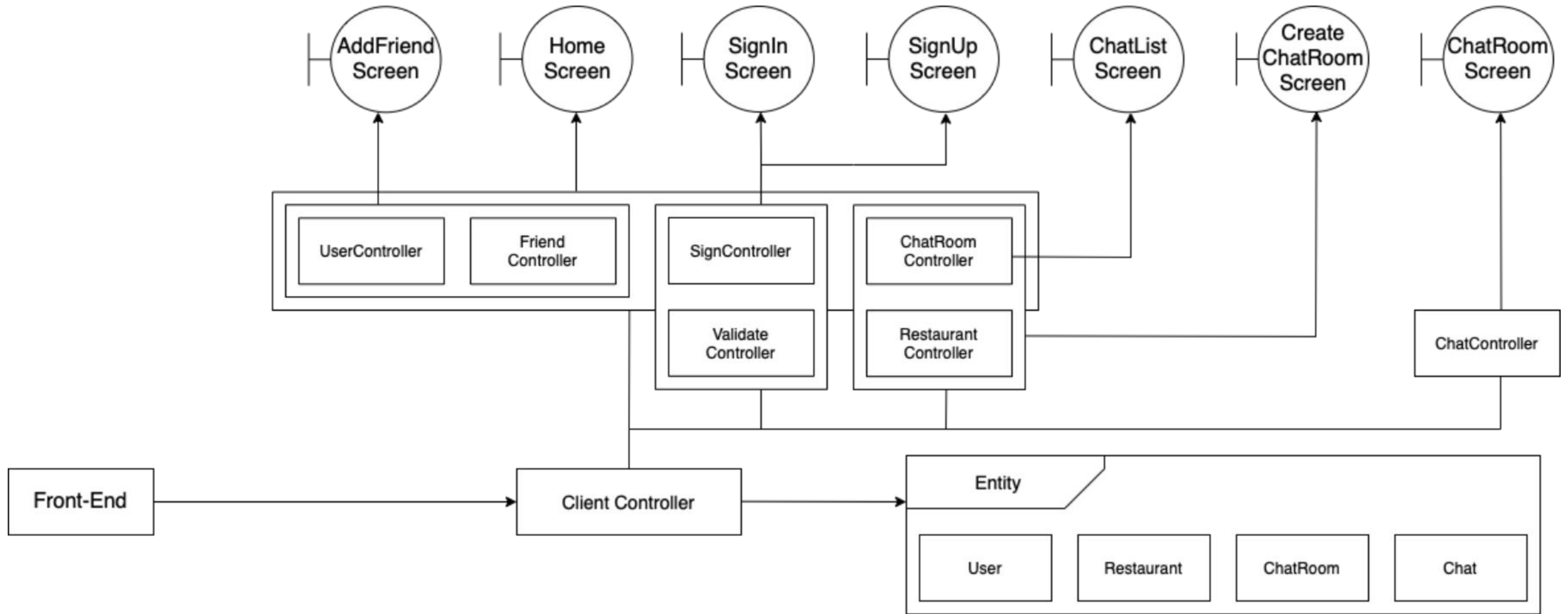
03 Sequence diagram 01. 다이어그램

채팅방 생성



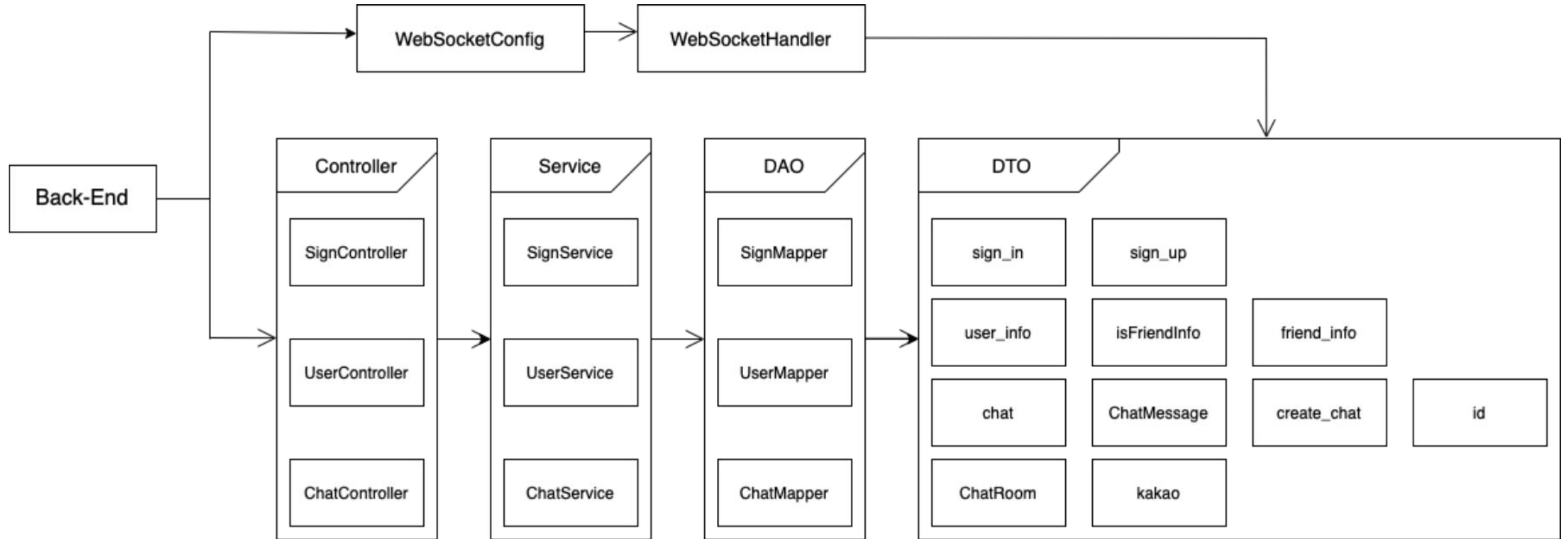
04 Class diagram

01. 다이어그램 02. 각 클래스 별 설명



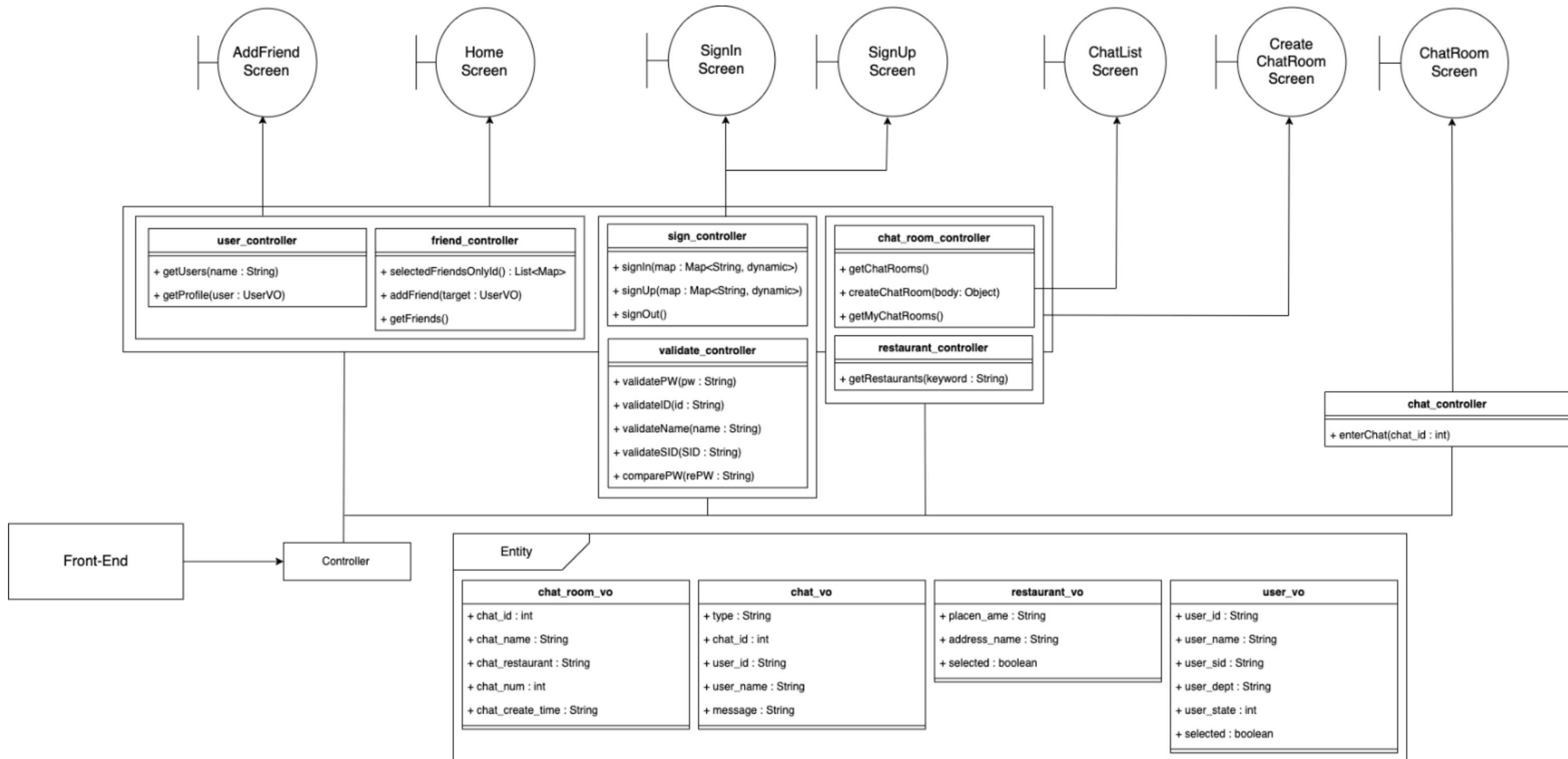
04 Class diagram

01. 다이어그램 02. 각 클래스 별 설명



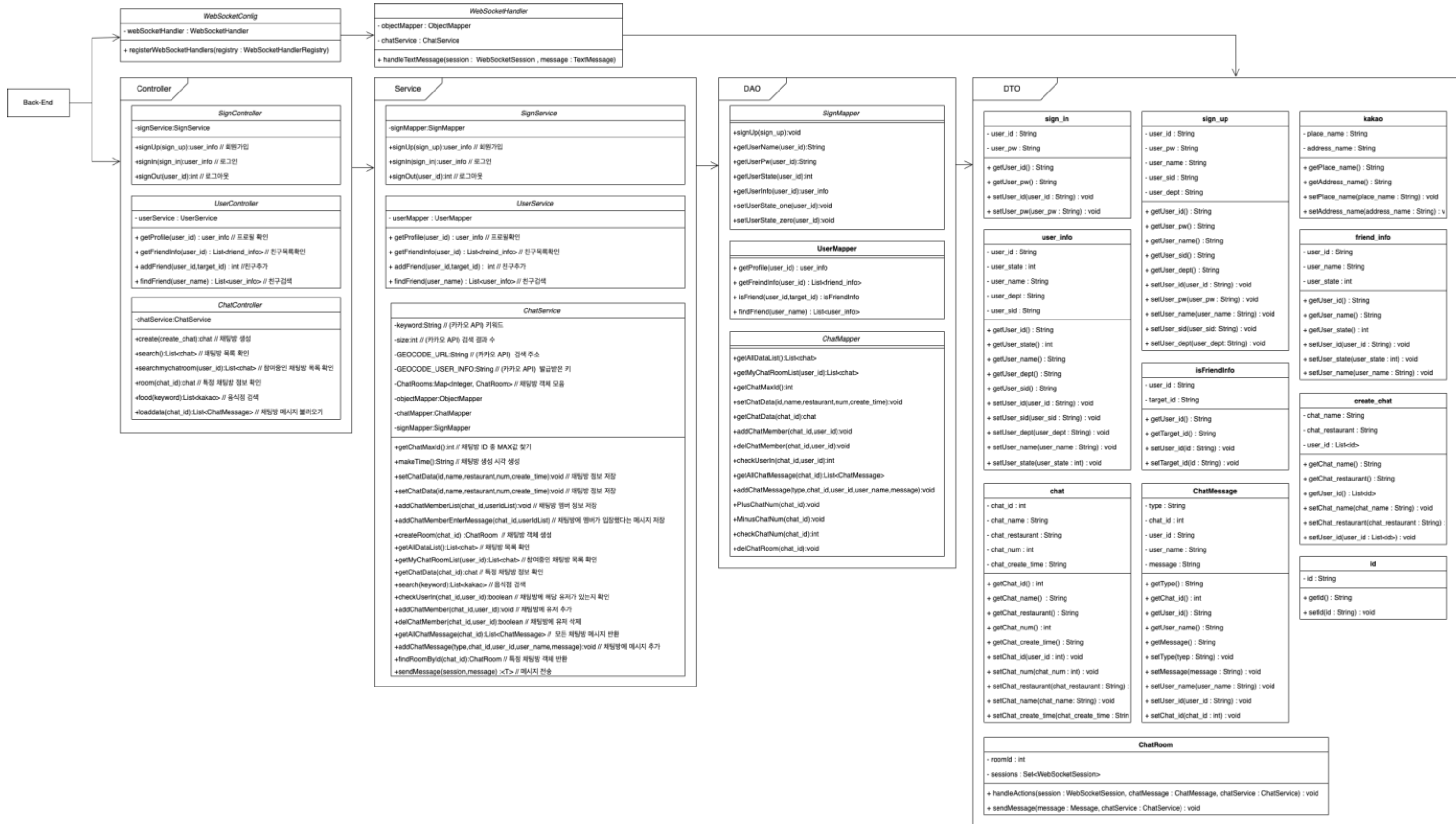
04 Class diagram

01. 다이어그램 02. 각 클래스 별 설명



04 Class diagram

01. 다이어그램 02. 각 클래스 별 설명



04 Class diagram

01. 다이어그램 02. 각 클래스 별 설명

<i>WebSocketConfig</i>
- webSocketHandler : WebSocketHandler
+ registerWebSocketHandlers(registry : WebSocketHandlerRegistry)

클래스명	WebSocketConfig	
클래스 개요	웹소켓 uri 매핑 클래스	
필드	webSocketHandler	웹소켓핸들러
메소드	registerWebSocketHandlers (registry)	웹소켓 uri 매핑 메소드

04 Class diagram

01. 다이어그램 02. 각 클래스 별 설명

<i>WebSocketHandler</i>
- objectMapper : ObjectMapper
- chatService : ChatService
+ handleMessage(session : WebSocketSession , message : TextMessage)

클래스명	WebSocketHandler	
클래스 개요	웹소켓 메세지 처리하는 클래스	
필드	objectMapper	objectmapper
	chatService	채팅방 메세지 내용 처리를 위한 객체
메소드	handleTextMessage(session, message)	세션별 메세지 처리 메소드

04 Class diagram

01. 다이어그램 02. 각 클래스 별 설명

<i>SignController</i>
-signService:SignService
+signUp(sign_up):user_info // 회원가입
+signIn(sign_in):user_info // 로그인
+signOut(user_id):int // 로그아웃

클래스명	SignController	
클래스 개요	로그인 로그아웃 회원가입을 처리하는 컨트롤러	
필드	signService	sign관련 메소드가 명시되어 있는 객체
메소드	signUP(sign_up)	회원가입
	signIn(sign_in)	로그인
	signOut(user_id)	로그아웃

04 Class diagram

01. 다이어그램 02. 각 클래스 별 설명

<i>UserController</i>
- userService : UserService
+ getProfile(user_id) : user_info // 프로필 확인
+ getFriendInfo(user_id) : List<friend_info> // 친구목록확인
+ addFriend(user_id,target_id) : int //친구추가
+ findFriend(user_name) : List<user_info> // 친구검색

클래스명	UserController	
클래스 개요	프로필확인 친구목록확인 친구추가 친구검색을 처리하는 컨트롤러	
필드	userService	사용자관련 메소드가 명시되어 있는 객체
메소드	getProfile(user_id)	프로필확인
	getFriendInfo(user_id)	친구목록확인
	addFriend(user_id, target_id)	친구추가
	findFriend(user_name)	친구검색

04 Class diagram

01. 다이어그램 02. 각 클래스 별 설명

<i>ChatController</i>
-chatService:ChatService
+create(create_chat):chat // 채팅방 생성
+search():List<chat> // 채팅방 목록 확인
+searchmychatroom(user_id):List<chat> // 참여중인 채팅방 목록 확인
+room(chat_id):chat // 특정 채팅방 정보 확인
+food(keyword):List<kakao> // 음식점 검색
+loaddata(chat_id):List<ChatMessage> // 채팅방 메시지 불러오기

클래스명	ChatController	
클래스 개요	채팅방 생성 채팅방 목록확인 참여중인채팅방 목록확인 특정 채팅방 정보확인 음식점 검색 채팅방 메시지 불러오기를 처리하는 컨트롤러	
필드	chatService	채팅방관련 메소드가 명시되어 있는 객체
메소드	create(create_chat)	채팅방 생성
	search()	채팅방 목록 확인
	searchmychatroom(user_id)	참여중인 채팅방 목록 확인
	room(chat_id)	특정 채팅방 정보 확인
	food(keyword)	음식점 검색
	loaddata(chat_id)	채팅방 메시지 불러오기

04 Class diagram

01. 다이어그램 02. 각 클래스 별 설명

<i>SignService</i>
-signMapper:SignMapper
+signUp(sign_up):user_info // 회원가입
+signIn(sign_in):user_info // 로그인
+signOut(user_id):int // 로그아웃

클래스명	SignService	
클래스 개요	회원가입 로그인 로그아웃 관련 메소드를 수행하는 서비스 클래스	
필드	signMapper	sign관련 쿼리가 수행되는 mapper
메소드	signUP(sign_up)	회원가입
	signIn(sign_in)	로그인
	signOut(user_id)	로그아웃

04 Class diagram

01. 다이어그램 02. 각 클래스 별 설명

<i>UserService</i>
- userMapper : UserMapper
+ getProfile(user_id) : user_info // 프로필확인
+ getFriendInfo(user_id) : List<freind_info> // 친구목록확인
+ addFriend(user_id,target_id) : int // 친구추가
+ findFriend(user_name) : List<user_info> // 친구검색

클래스명	UserService	
클래스 개요	프로필확인 친구목록확인 친구추가 친구검색 메소드를 수행하는 서비스 클래스	
필드	userMapper	사용자관련 쿼리가 수행되는 mapper
메소드	getProfile(user_id)	프로필확인
	getFriendInfo(user_id)	친구목록확인
	addFriend(user_id, target_id)	친구추가
	findFriend(user_name)	친구검색

04 Class diagram

01. 다이어그램 02. 각 클래스 별 설명

```

class ChatService {
    -keyword:String // (카카오 API) 키워드
    -size:int // (카카오 API) 검색 결과 수
    -GEOCODE_URL:String // (카카오 API) 검색 주소
    -GEOCODE_USER_INFO:String // (카카오 API) 발급받은 키
    -ChatRooms:Map<Integer, ChatRoom> // 채팅방 객체 모음
    -objectMapper:ObjectMapper
    -chatMapper:ChatMapper
    -signMapper:SignMapper

    +getChatMaxId():int // 채팅방 ID 중 MAX값 찾기
    +makeTime():String // 채팅방 생성 시각 생성
    +setChatData(id,name,restaurant,num,create_time):void // 채팅방 정보 저장
    +setChatData(id,name,restaurant,num,create_time):void // 채팅방 정보 저장
    +addChatMemberList(chat_id,userIdList):void // 채팅방 멤버 정보 저장
    +addChatMemberEnterMessage(chat_id,userIdList) // 채팅방에 멤버가 입장했다는 메시지 저장
    +createRoom(chat_id) :ChatRoom // 채팅방 객체 생성
    +getAllDataList():List<chat> // 채팅방 목록 확인
    +getMyChatRoomList(user_id):List<chat> // 참여중인 채팅방 목록 확인
    +getChatData(chat_id):chat // 특정 채팅방 정보 확인
    +search(keyword):List<kakao> // 음식점 검색
    +checkUserIn(chat_id,user_id):boolean // 채팅방에 해당 유저가 있는지 확인
    +addChatMember(chat_id,user_id):void // 채팅방에 유저 추가
    +delChatMember(chat_id,user_id):boolean // 채팅방에 유저 삭제
    +getAllChatMessage(chat_id):List<ChatMessage> // 모든 채팅방 메시지 반환
    +addChatMessage(type,chat_id,user_id,user_name,message):void // 채팅방에 메시지 추가
    +findRoomById(chat_id):ChatRoom // 특정 채팅방 객체 반환
    +sendMessage(session,message) :<T> // 메시지 전송
}
    
```

클래스명	ChatService	
클래스 개요	채팅방 관련 메소드를 수행하는 서비스 클래스	
필드	keyword	(카카오API)키워드
	size	(카카오API)검색 결과 수
	GEOCODE_URL	(카카오API)검색 주소
	GEOCODE_USER_INFO	(카카오API)발급받은 키
	ChatRooms	채팅방 객체 모음
	objectMapper	오브젝트 mapper
	chatMapper	채팅방 mapper
	signMapper	signmapper

04 Class diagram

01. 다이어그램 02. 각 클래스 별 설명

```

class ChatService {
    -keyword:String // (카카오 API) 키워드
    -size:int // (카카오 API) 검색 결과 수
    -GEOCODE_URL:String // (카카오 API) 검색 주소
    -GEOCODE_USER_INFO:String // (카카오 API) 발급받은 키
    -ChatRooms:Map<Integer, ChatRoom> // 채팅방 객체 모음
    -objectMapper:ObjectMapper
    -chatMapper:ChatMapper
    -signMapper:SignMapper

    +getChatMaxId():int // 채팅방 ID 중 MAX값 찾기
    +makeTime():String // 채팅방 생성 시각 생성
    +setChatData(id,name,restaurant,num,create_time):void // 채팅방 정보 저장
    +setChatData(id,name,restaurant,num,create_time):void // 채팅방 정보 저장
    +addChatMemberList(chat_id,userIdList):void // 채팅방 멤버 정보 저장
    +addChatMemberEnterMessage(chat_id,userIdList) // 채팅방에 멤버가 입장했다는 메시지 저장
    +createRoom(chat_id) :ChatRoom // 채팅방 객체 생성
    +getAllDataList():List<chat> // 채팅방 목록 확인
    +getMyChatRoomList(user_id):List<chat> // 참여중인 채팅방 목록 확인
    +getChatData(chat_id):chat // 특정 채팅방 정보 확인
    +search(keyword):List<kakao> // 음식점 검색
    +checkUserIn(chat_id,user_id):boolean // 채팅방에 해당 유저가 있는지 확인
    +addChatMember(chat_id,user_id):void // 채팅방에 유저 추가
    +delChatMember(chat_id,user_id):boolean // 채팅방에 유저 삭제
    +getAllChatMessage(chat_id):List<ChatMessage> // 모든 채팅방 메시지 반환
    +addChatMessage(type,chat_id,user_id,user_name,message):void // 채팅방에 메시지 추가
    +findRoomById(chat_id):ChatRoom // 특정 채팅방 객체 반환
    +sendMessage(session,message) :<T> // 메시지 전송
}
    
```

클래스명	ChatService	
클래스 개요	채팅방 관련 메소드를 수행하는 서비스 클래스	
메소드	getChatMaxId()	채팅방 ID중 MAX값 찾기
	makeTime()	채팅방 생성 시각 생성
	setChatData(id, name, restaurant, num, create_time)	채팅방 정보 저장
	addChatMemberList(chat_id, userIdList)	채팅방 멤버 정보 저장
	addChatMemberEnterMessage(chat_id, userIdList)	태이채팅방에 멤버가 입장했다는 메시지 저장
	createRoom(chat_id)	채팅방 생성
	getAllDataList()	채팅방 목록 확인
	getMyChatRoomList(user_id)	참여중인 채팅방 목록 확인
	getChatData(chat_id)	특정 채팅방 정보 확인
	search(keyword)	음식점 검색
	checkUserIn(chat_id, user_id)	채팅방에 해당 유저가 있는지 확인
	addChatMember(chat_id, user_id)	채팅방에 유저 추가
	delChatMember(chat_id, user_id)	채팅방에 유저 삭제
	getAllChatMessage(chat_id)	모든 채팅방 메시지 반환
	addChatMessage(type, chat_id, user_id, user_name, message)	채팅방에 메시지 추가
findRoomById(chat_id)	특정 채팅방 객체 반환	
sendMessage(session, message)	메시지 전송	

04 Class diagram

01. 다이어그램 02. 각 클래스 별 설명

<i>SignMapper</i>
+signUp(sign_up):void +getUserName(user_id):String +getUserPw(user_id):String +getUserState(user_id):int +getUserInfo(user_id):user_info +setUserState_one(user_id):void +setUserState_zero(user_id):void

클래스명	SignMapper	
클래스 개요	sign관련 table 연결되어있는 mapper 클래스	
메소드	signUp(sign_up)	회원가입
	getUserName(user_id)	유저이름 반환
	getUserPW(user_id)	유저비밀번호 반환
	getUserstate(user_id)	유저현재상태 반환
	getUserInfo(user_id)	유저에 대한 정보 반환
	setUserInfo(user_id)	유저에 대한 정보 입력
	setUserState_one(user_id)	유저 온라인 상태로 변경
	setUserState_zero(user_id)	유저 오프라인 상태로 변경

04 Class diagram

01. 다이어그램 02. 각 클래스 별 설명

UserMapper
+ getProfile(user_id) : user_info + getFreindInfo(user_id) : List<friend_info> + isFriend(user_id,target_id) : isFriendInfo + findFriend(user_name) : List<user_info>

클래스명	UserMapper	
클래스 개요	사용자관련 table 연결되어있는 mapper 클래스	
메소드	getProfile(user_id)	사용자 프로필 반환
	getFriendInfo(user_id)	사용자의 친구목록 반환
	isFriend(user_id, target_id)	친구 유무 확인
	findFriend(user_name)	이름에 해당하는 친구 검색

04 Class diagram

01. 다이어그램 02. 각 클래스 별 설명

ChatMapper
+getAllDataList():List<chat>
+getMyChatRoomList(user_id):List<chat>
+getChatMaxId():int
+setChatData(id,name,restaurant,num,create_time):void
+getChatData(chat_id):chat
+addChatMember(chat_id,user_id):void
+delChatMember(chat_id,user_id):void
+checkUserIn(chat_id,user_id):int
+getAllChatMessage(chat_id):List<ChatMessage>
+addChatMessage(type,chat_id,user_id,user_name,message):void
+PlusChatNum(chat_id):void
+MinusChatNum(chat_id):void
+checkChatNum(chat_id):int
+delChatRoom(chat_id):void

클래스명	ChatService	
클래스 개요	채팅방 관련 table 연결되어있는 mapper 클래스	
메소드	getAllDataList()	모든 채팅방 목록 반환
	getMyChatRoomList(user_id)	사용자가 들어있는 채팅방 목록 반환
	getChatMaxId()	채팅방 아이디 중 가장 큰 값 반환
	setChatDate(id, name, restaurant, num, create_time)	채팅방 생성
	getChatData(chat_id)	채팅방 정보 반환
	addChatMember(chat_id, user_id)	채팅방에 사용자 추가
	delChatMember(chat_id, user_id)	채팅방에서 사용자 삭제
	checkUserIn(chat_id, user_id)	채팅방에 유저가 들어있는지 확인
	getAllChatMessage(chat_id)	채팅방의 메시지 반환
	addChatMessage(type, chat_id, user_id, user_name, message)	채팅방 메시지 입력
	PlusChatNum(chat_id)	채팅방 인원 수 추가
	MinusChatNum(chat_id)	채팅방 인원 수 감소
	checkChatNum(chat_id)	채팅방 인원 수 확인
	delChatRoom(chat_id)	채팅방 삭제

04 Class diagram

01. 다이어그램 02. 각 클래스 별 설명

sign_in
- user_id : String - user_pw : String
+ getUser_id() : String + getUser_pw() : String + setUser_id(user_id : String) : void + setUser_pw(user_pw : String) : void

클래스명	sign_in	
클래스 개요	로그인에 필요한 정보를 보관	
필드	user_id : String	사용자 아이디
	user_pw : String	사용자 비밀번호 (해시값)
메소드	getUser_id() : String	사용자 아이디 반환
	getUser_pw() : String	사용자 비밀번호 반환
	setUser_id(user_id : String) : void	사용자 아이디 설정
	setUser_pw(user_pw : String) : void	사용자 비밀번호 설정

04 Class diagram

01. 다이어그램

02. 각 클래스 별 설명

user_info
- user_id : String
- user_state : int
- user_name : String
- user_dept : String
- user_sid : String
+ getUser_id() : String
+ getUser_state() : int
+ getUser_name() : String
+ getUser_dept() : String
+ getUser_sid() : String
+ setUser_id(user_id : String) : void
+ setUser_sid(user_sid : String) : void
+ setUser_dept(user_dept : String) : void
+ setUser_name(user_name : String) : void
+ setUser_state(user_state : int) : void

클래스명	user_info	
클래스 개요	회원 정보를 보관	
필드	user_id : String	사용자 아이디
	user_state : int	사용자 온/오프라인 상태
	user_name : String	사용자 이름
	user_dept : String	사용자 학과
	user_sid : String	사용자 학번
메소드	getUser_id() : String	사용자 아이디 반환
	getUser_state() : int	사용자 온/오프라인 상태 반환
	getUser_name() : String	사용자 이름 반환
	getUser_dept() : String	사용자 학과 반환
	getUser_sid() : String	사용자 학번 반환
	setUser_id(user_id : String) : void	사용자 아이디 설정
	setUser_state(user_state : int) : void	사용자 온/오프라인 상태 설정
	setUser_name(user_name : String) : void	사용자 이름 설정
	setUser_dept(user_dept : String) : void	사용자 학과 설정
	setUser_sid(user_sid : String) : void	사용자 학번 설정

04 Class diagram

01. 다이어그램 02. 각 클래스 별 설명

sign_up
- user_id : String
- user_pw : String
- user_name : String
- user_sid : String
- user_dept : String
+ getUser_id() : String
+ getUser_pw() : String
+ getUser_name() : String
+ getUser_sid() : String
+ getUser_dept() : String
+ setUser_id(user_id : String) : void
+ setUser_pw(user_pw : String) : void
+ setUser_name(user_name : String) : void
+ setUser_sid(user_sid: String) : void
+ setUser_dept(user_dept: String) : void

클래스명	sign_up	
클래스 개요	회원가입에 필요한 정보를 보관	
필드	user_id : String	사용자 아이디
	user_pw : String	사용자 비밀번호 (해시값)
	user_name : String	사용자 이름
	user_sid : String	사용자 학번
	user_dept : String	사용자 학과
메소드	getUser_id(): String	사용자 아이디 반환
	getUser_pw(): String	사용자 비밀번호 반환
	getUser_name(): String	사용자 이름 반환
	getUser_sid(): String	사용자 학번 반환
	getUser_dept(): String	사용자 학과 반환
	setUser_id(user_id : String) : void	사용자 아이디 설정
	setUser_pw(user_pw : String) : void	사용자 비밀번호 설정
	setUser_name(user_name : String) : void	사용자 이름 설정
	setUser_sid(user_sid: String) : void	사용자 학번 설정
	setUser_dept(user_dept: String) : void	사용자 학과 설정

04 Class diagram

01. 다이어그램 02. 각 클래스 별 설명

isFriendInfo
- user_id : String
- target_id : String
+ getUser_id() : String
+ getTarget_id() : String
+ setUser_id(id : String) : void
+ setTarget_id(id : String) : void

클래스명	isFriendInfo	
클래스 개요	친구 추가에 필요한 정보를 보관	
필드	user_id : String	사용자 아이디
	target_id : String	친구 초대한 아이디
메소드	getUser_id(): String	사용자 아이디 반환
	getTarget_id(): String	친구 초대한 아이디 반환
	setUser_id(id : String) : void	사용자 아이디 설정
	setTarget_id(id : String) : void	친구 초대한 아이디 설정

04 Class diagram

01. 다이어그램 02. 각 클래스 별 설명

friend_info
- user_id : String
- user_name : String
- user_state : int
+ getUser_id() : String
+ getUser_name() : String
+ getUser_state() : int
+ setUser_id(user_id : String) : void
+ setUser_state(user_state : int) : void
+ setUser_name(user_name : String) : void

클래스명	friend_info	
클래스 개요	친구 정보를 보관	
필드	user_id : String	사용자 아이디
	user_name : String	사용자 이름
	user_state : int	사용자 온/오프라인 상태
메소드	getUser_id() : String	사용자 아이디 반환
	getUser_name() : String	사용자 이름 반환
	getUser_state() : int	사용자 온/오프라인 상태 반환
	setUser_id(user_id : String) : void	사용자 아이디 설정
	setUser_state(user_state : int) : void	사용자 온/오프라인 상태 설정
	setUser_name(user_name : String) : void	사용자 이름 설정

04 Class diagram

01. 다이어그램 02. 각 클래스 별 설명

create_chat
- chat_name : String
- chat_restaurant : String
- user_id : List<id>
+ getChat_name() : String
+ getChat_restaurant() : String
+ getUser_id() : List<id>
+ setChat_name(chat_name : String) : void
+ setChat_restaurant(chat_restaurant : String) : void
+ setUser_id(user_id : List<id>) : void

클래스명	create_chat	
클래스 개요	채팅방 생성에 필요한 정보를 보관	
필드	chat_name : String	채팅방 이름
	chat_restaurant : String	채팅방 음식점
	user_id : List<id>	입장할 사용자 목록
메소드	getChat_name() : String	채팅방 이름 반환
	getChat_restaurant() : String	채팅방 음식점 반환
	getUser_id() : List<id>	입장할 사용자 목록 반환
	setChat_name(chat_name : String) : void	채팅방 이름 설정
	setChat_restaurant(chat_restaurant : String) : void	채팅방 음식점 설정
	setUser_id(user_id : List<id>) : void	입장할 사용자 목록 설정

04 Class diagram

01. 다이어그램 02. 각 클래스 별 설명

ChatRoom
- roomId : int - sessions : Set<WebSocketSession>
+ handleActions(session : WebSocketSession, chatMessage : ChatMessage, chatService : ChatService) : void + sendMessage(message : Message, chatService : ChatService) : void

클래스명	ChatRoom	
클래스 개요	채팅방 제어 (입장, 대화, 퇴장, 뒤로가기)	
필드	roomId : int	채팅방 아이디
	sessions : Set<WebSocketSession>	세션 목록
메소드	handleActions(session : WebSocketSession, chatMessage : ChatMessage, chatService : ChatService) : void	채팅방 제어 (입장, 대화, 퇴장, 뒤로가기)
	sendMessage(message : Message, chatService : ChatService) : void	메시지 전송

04 Class diagram

01. 다이어그램 02. 각 클래스 별 설명

id
- id : String
+ getId() : String
+ setId(id : String) : void

클래스명	id	
클래스 개요	사용자 아이디를 보관	
필드	id : String	사용자 아이디
메소드	getId() : String	사용자 아이디 반환
	setId(id : String) : void	사용자 아이디 설정

04 Class diagram

01. 다이어그램 02. 각 클래스 별 설명

chat
- chat_id : int - chat_name : String - chat_restaurant : String - chat_num : int - chat_create_time : String
+ getChat_id() : int + getChat_name() : String + getChat_restaurant() : String + getChat_num() : int + getChat_create_time() : String + setChat_id(user_id : int) : void + setChat_num(chat_num : int) : void + setChat_restaurant(chat_restaurant : String) : void + setChat_name(chat_name: String) : void + setChat_create_time(chat_create_time : String) : void

클래스명	chat	
클래스 개요	채팅방 정보를 보관	
필드	chat_id : int	채팅방 아이디
	chat_name : String	채팅방 이름
	chat_restaurant : String	채팅방 음식점
	chat_num : int	채팅방 인원 수
	chat_create_time : String	채팅방 생성 시각
메소드	getChat_id() : int	채팅방 아이디 반환
	getChat_name() : String	채팅방 이름 반환
	getChat_restaurant() : String	채팅방 음식점 반환
	getChat_num() : int	채팅방 인원 수 반환
	getChat_create_time() : String	채팅방 생성 시각 반환
	setChat_id(user_id : int) : void	채팅방 아이디 설정
	setChat_name(chat_name: String) : void	채팅방 이름 설정
	setChat_restaurant(chat_restaurant : String) : void	채팅방 인원 수 설정
	setChat_num(chat_num : int) : void	채팅방 생성 시각 설정
	setChat_create_time(chat_create_time : String) : void	채팅방 생성 시각 설정

04 Class diagram

01. 다이어그램 02. 각 클래스 별 설명

ChatMessage
- type : String - chat_id : int - user_id : String - user_name : String - message : String
+ getType() : String + getChat_id() : int + getUser_id() : String + getUser_name() : String + getMessage() : String + setType(tyep : String) : void + setMessage(message : String) : void + setUser_name(user_name : String) : void + setUser_id(user_id : String) : void + setChat_id(chat_id : int) : void

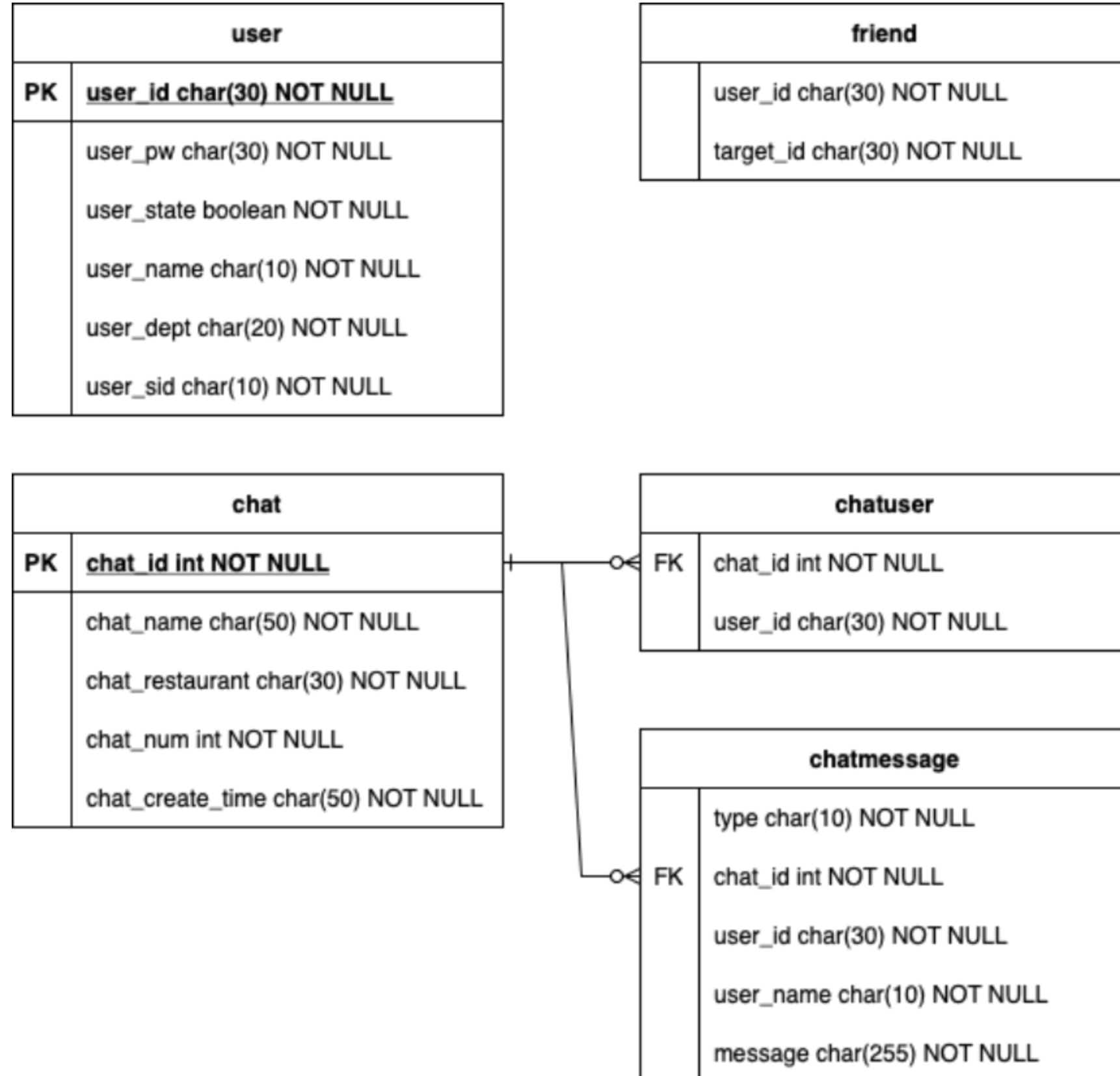
클래스명	ChatMessage	
클래스 개요	채팅방 메시지 정보를 보관	
필드	type : String	메시지 타입
	chat_id : int	채팅방 아이디
	user_id : String	사용자 아이디
	user_name : String	사용자 이름
	message : String	메시지 내용
메소드	getType() : String	메시지 타입 반환
	getChat_id() : int	채팅방 아이디 반환
	getUser_id() : String	사용자 아이디 반환
	getUser_name() : String	사용자 이름 반환
	getMessage() : String	메시지 내용 반환
	setType(tyep : String) : void	메시지 타입 설정
	setChat_id(chat_id : int) : void	채팅방 아이디 설정
	setUser_id(user_id : String) : void	사용자 아이디 설정
	setUser_name(user_name : String) : void	사용자 이름 설정
	setMessage(message : String) : void	메시지 내용 설정

04 Class diagram

01. 다이어그램 02. 각 클래스 별 설명

kakao
- place_name : String
- address_name : String
+ getPlace_name() : String
+ getAddress_name() : String
+ setPlace_name(place_name : String) : void
+ setAddress_name(address_name : String) : void

클래스명	kakao	
클래스 개요	카카오 API로 음식점을 검색한 결과를 보관	
필드	place_name : String	음식점 이름
	address_name : String	음식점 주소
메소드	getPlace_name() : String	음식점 이름 반환
	getAddress_name() : String	음식점 주소 반환
	setPlace_name(place_name : String) : void	음식점 이름 설정
	setAddress_name(address_name : String) : void	음식점 주소 설정



WE_EAT Database

1. User Table

- 회원의 정보가 담긴 테이블
- 회원의 ID를 기본 키로 가짐

2. Friend Table

- 친구 관계 정보가 담긴 테이블
- [친구 요청을 한 사용자 ID, 친구 요청을 받은 사용자 ID]로 구성됨

3. Chat Table

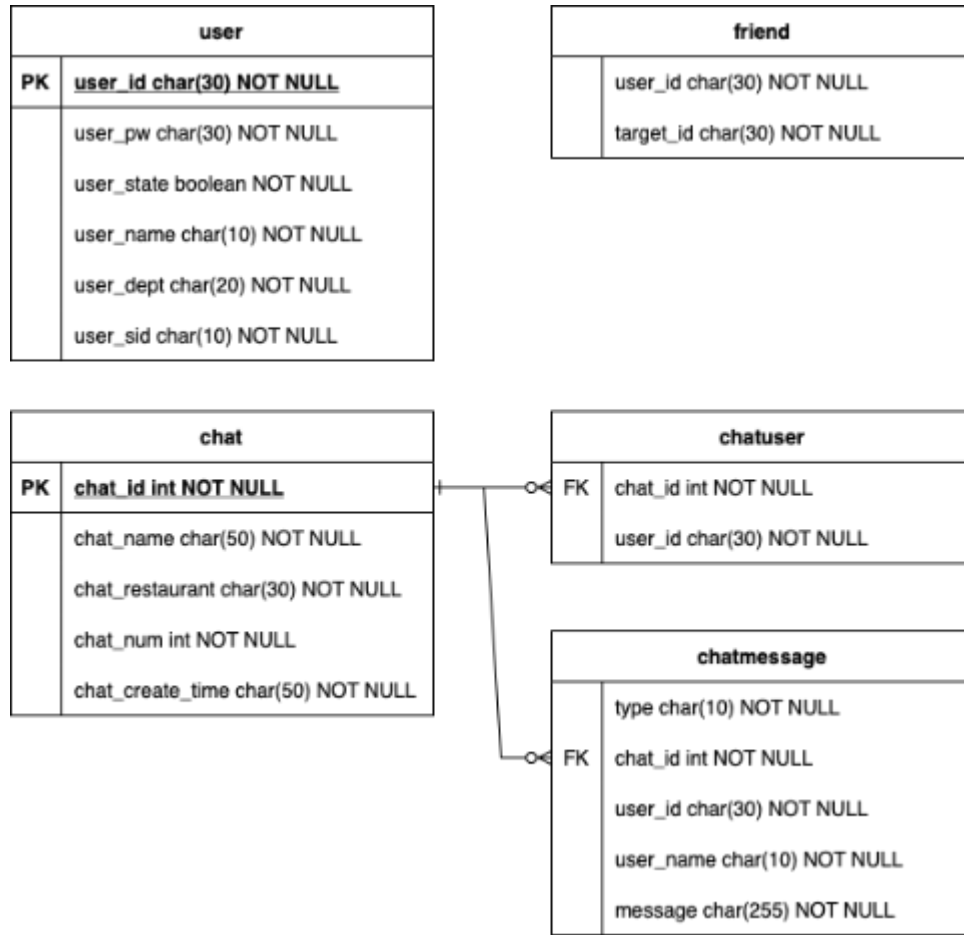
- 채팅방 정보가 담긴 테이블
- 채팅방의 ID를 기본 키로 가짐

4. ChatUser Table

- 채팅방 참여 정보가 담긴 테이블
- [채팅방 ID, 채팅방에 참가한 사용자 ID]로 구성됨
- Chat Table의 기본 키(채팅방 ID)를 외래 키로 가짐(cascade option)

5. Chatmessage Table

- 채팅방 메시지 정보가 담긴 테이블
- [채팅 type, 채팅방 ID, 사용자 ID, 사용자 이름, 메시지]로 구성됨
- Chat Table의 기본 키(채팅방 ID)를 외래 키로 가짐(cascade option)



테이블명	설명	유즈케이스명	유즈케이스 ID
User	회원 정보 테이블	회원가입 로그인 로그아웃 친구상태실시간확인 프로필확인	WE_SignUp WE_SignIn WE_SignOut WE_CheckStatus WE_CheckProfile
Friend	친구 관계 정보 테이블	친구추가 친구상태실시간확인 채팅방생성	WE_AddFriend WE_CheckStatus WE_CreateChatRoom
Chat	채팅방 정보 테이블	채팅방 생성 채팅방 목록 확인	WE_CreateChatRoom WE_CheckChatRoomList WE_ExitChatRoom
ChatUser	채팅방 참여 정보 테이블	채팅방 생성 채팅방 참여 채팅방 나가기	WE_CreateChatRoom WE_JoinChatRoom WE_ExitChatRoom
ChatMessage	채팅방 메시지 정보 테이블	1:N 채팅	WE_Chating

User Table					
user_id	user_pw	user_state	user_name	user_dept	user_sid
qwer1234	96cae35c8a9	1	박도윤	컴퓨터정보공학부	201720490
sha12345	bfd7f58f887b	1	안성현	컴퓨터정보공학부	201921725
sssm4444	c59b9fb723a	1	황주연	자유전공학과	202020202
user0283	8d969eef6ec	1	이승민	정보통신전자공학부	201621053
user2000	8d969eef6ec	1	한영찬	컴퓨터정보공학부	201821284

User Table

- 회원의 정보가 담긴 테이블

Chat Table				
chat_id	chat_name	chat_restaurant	chat_num	chat_create_time
1	먹기 위해 팀플하는 사람	굽네치킨 역곡점	4	22.06.11 5pm
2	초밥 드실 분	스시마리오	5	22.06.11 5pm
3	바나나 케이크 먹을 사람	바나나쉽 소사점	2	22.06.11 5pm

Chat Table

- 채팅방 정보가 담긴 테이블

Friend Table	
user_id	target_id
user0283	user2000
user2000	user0283
sssm4444	sha12345
sha12345	sssm4444
sssm4444	user2000
user2000	sssm4444

Friend Table

- 친구 관계 정보가 담긴 테이블

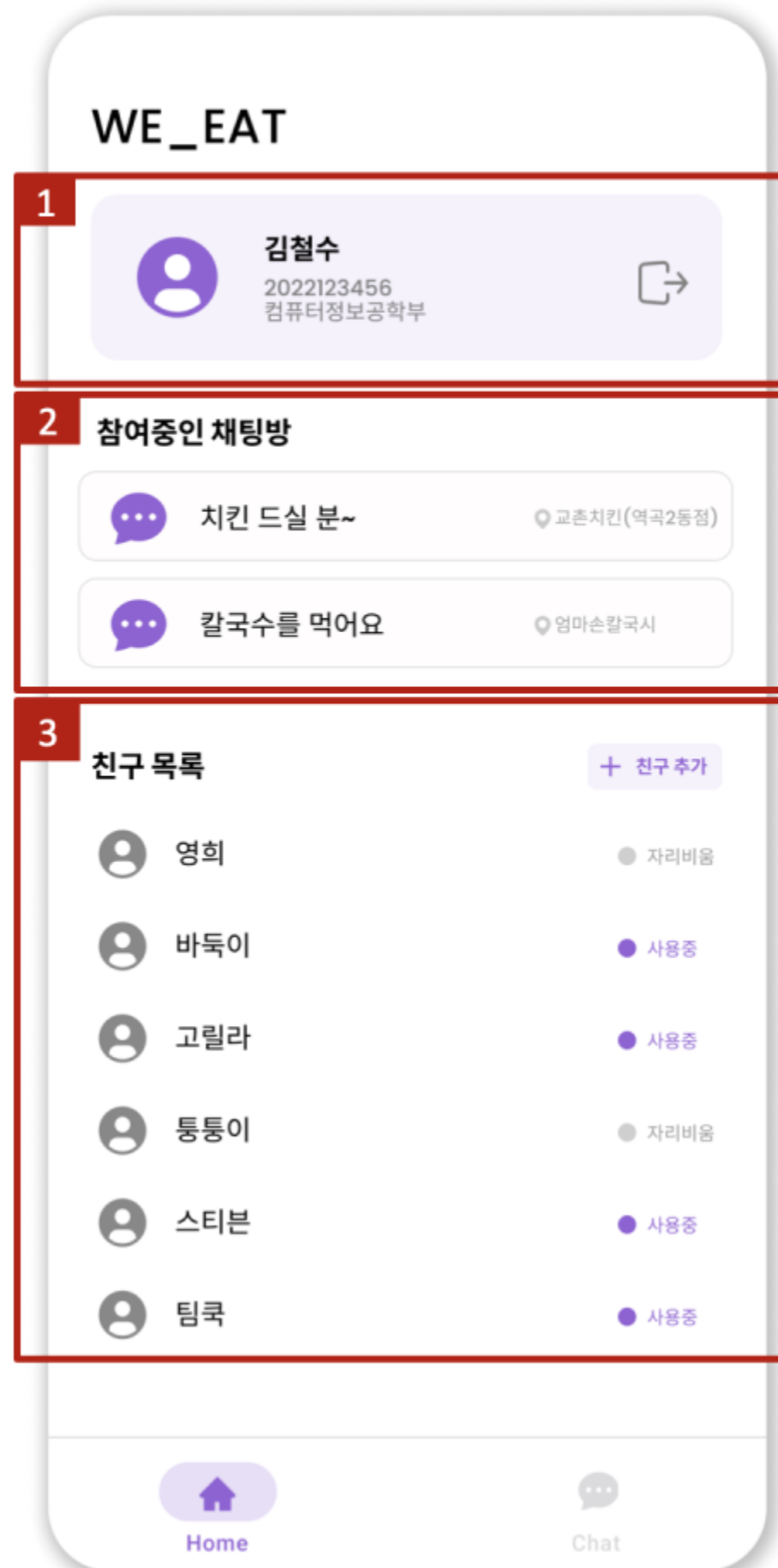
ChatUser Table	
chat_id	user_id
1	user2000
2	user0283
1	user0283
1	sssm4444
2	sha12345
2	qwer1234

ChatUser Table

- 채팅방 참여 정보가 담긴 테이블

ChatMessage Table				
type	chat_id	user_id	user_name	message
ENTER	1	user2000	한영찬	
ENTER	2	user0283	이승민	
ENTER	1	user0283	이승민	
TALK	1	user0283	이승민	안녕하세요~
TALK	1	user2000	한영찬	안녕하세요
OUT	2	user0283	이승민	

ChatMessage Table
- 채팅방 대화 내용이 담긴 테이블



홈 화면 - HomeScreen

하단 내비게이션 바의 HOME에 해당 하는 UI 앱 사용에 있어 기본 정보를 담고 있음

1. 사용자 정보 및 로그아웃

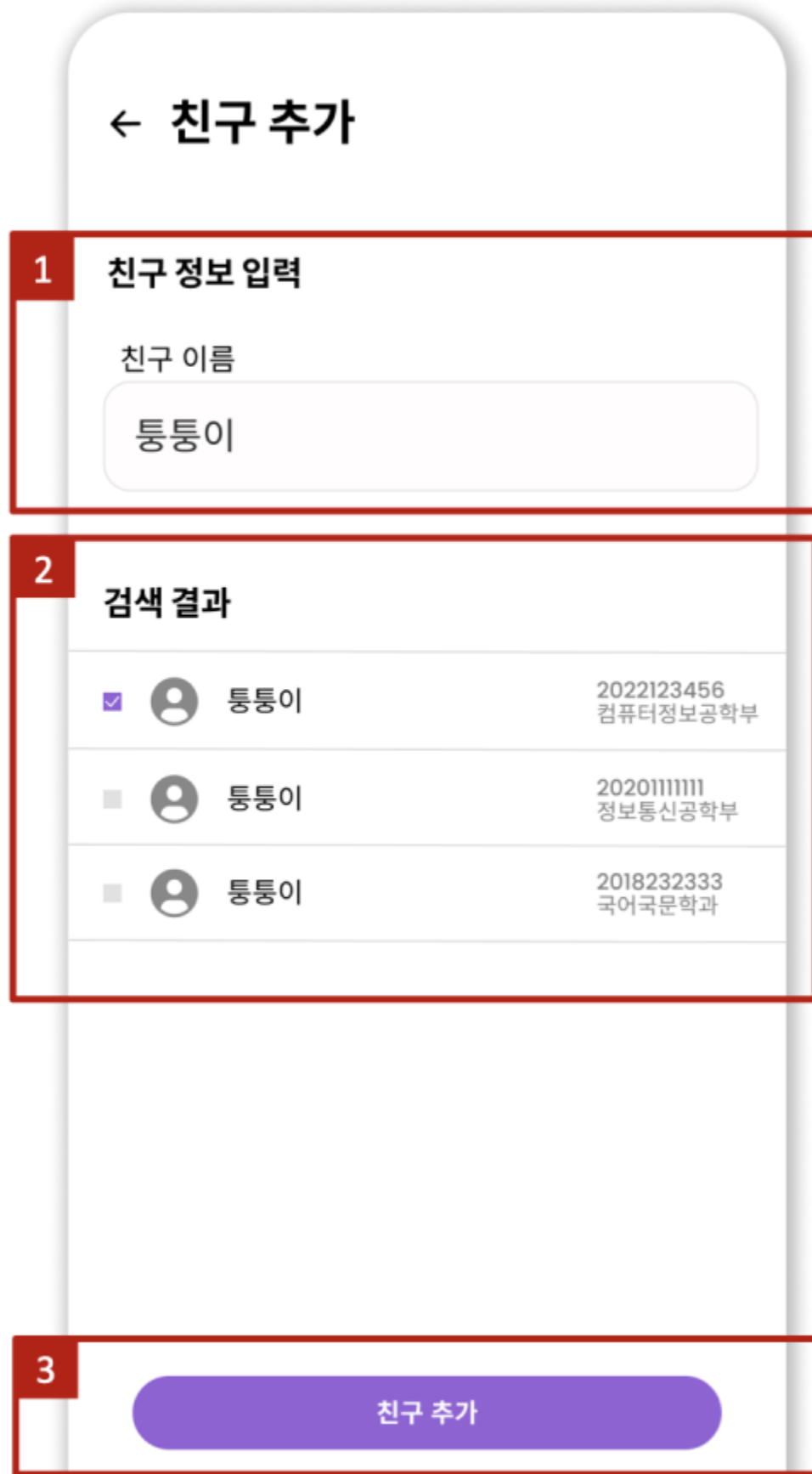
- 해당 사용자 정보 표시(이름, 학번, 전공)
- 우측 로그아웃 아이콘을 클릭하여 로그아웃 가능

2. 참여중인 채팅방 정보

- 현재 사용자가 참여중인 채팅방 정보 제공
- 클릭 시 해당 채팅방으로 이동

3. 친구 목록 정보 및 친구 추가

- 현재 사용자의 친구 목록 제공
- 친구들의 앱 사용 상태(자리비움, 사용중) 표시
- 친구 추가 버튼을 통해 친구 목록을 추가할 수 있는 화면으로 이동



친구 추가 화면 - AddFriendScreen

Home에서 친구 추가 버튼을 통해 접근할 수 있는 친구 추가 화면
이름 검색을 통해 친구를 추가할 수 있음

1. 친구 정보 입력란

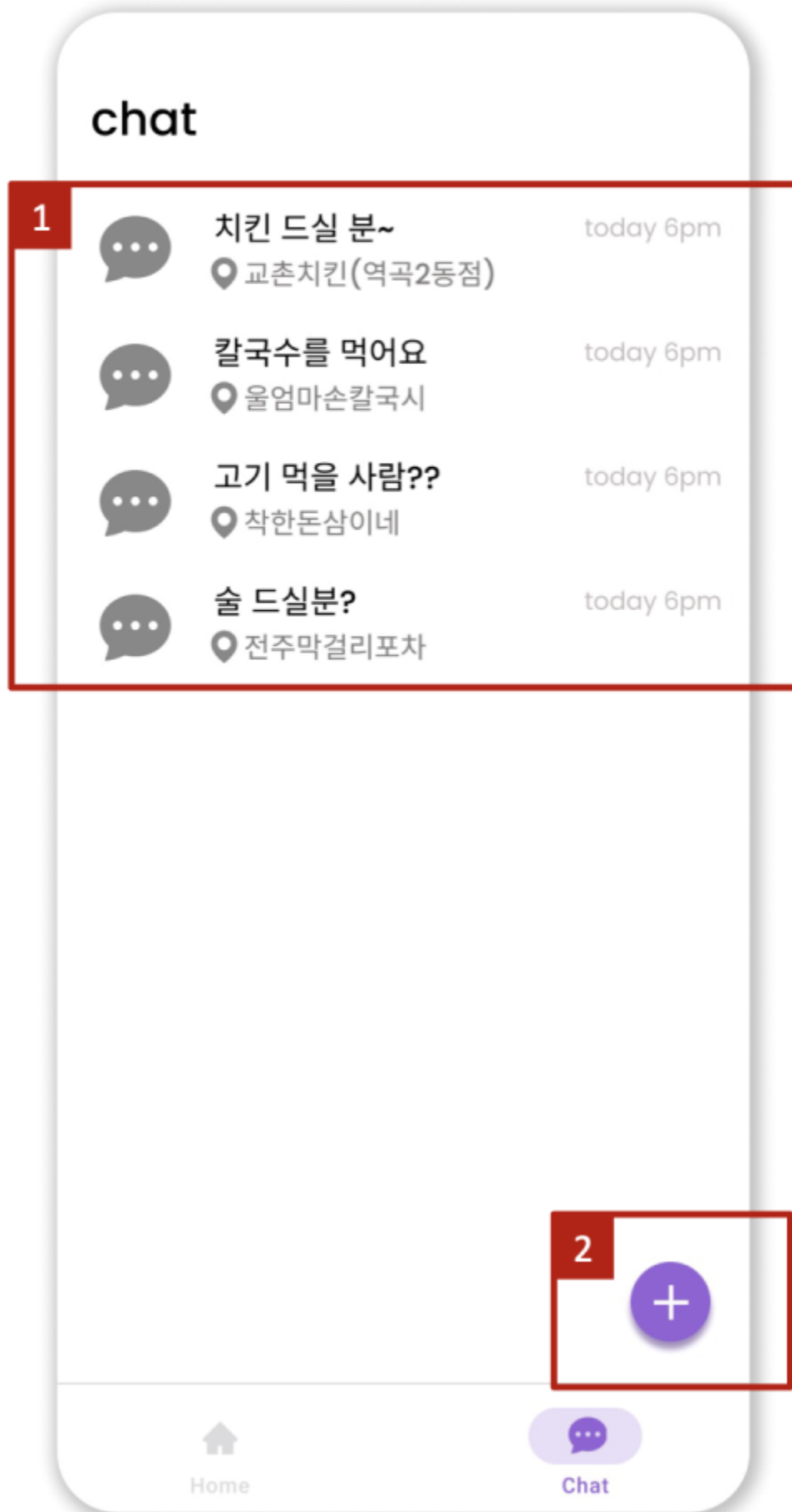
- 추가 하고자 하는 친구의 이름을 검색하는 검색창

2. 검색 결과

- 검색한 친구 이름을 기반으로 User DB에 저장된 사용자 정보를 출력
- 체크 버튼을 통해 친구 추가를 하고 싶은 사용자를 확정

3. 친구 추가 버튼

- 사용자 확정(체크)가 된 친구를 친구 목록에 추가하는 친구 추가 버튼



전체 채팅방 목록 화면 - ChatListScreen

하단 내비게이션 바의 chat 부분을 구성하는 화면
현재 생성된 전체 채팅방 목록을 보여줌
채팅방 생성 버튼을 통해 새로운 채팅방을 생성할 수 있음

1. 전체 채팅방 목록

- 현재 생성된 전체 채팅방 목록을 보여줌
- 해당 채팅방 클릭 시 채팅 화면으로 이동함

2. 새로운 채팅방 생성 버튼

- 새로운 채팅방을 생성할 수 있는 채팅방 생성 버튼
- 클릭 시 새로운 채팅방을 생성하는 채팅방 생성 화면으로 이동함

채팅방 생성 화면 – CreateChatRoomScreen

채팅방 생성 버튼 클릭 시 보여지는 화면
채팅방 기본 정보를 입력하고 채팅방을 생성할 수 있음

1. 채팅방 이름 설정

- 생성할 채팅방의 이름을 입력함

2. 음식점 검색 및 선택

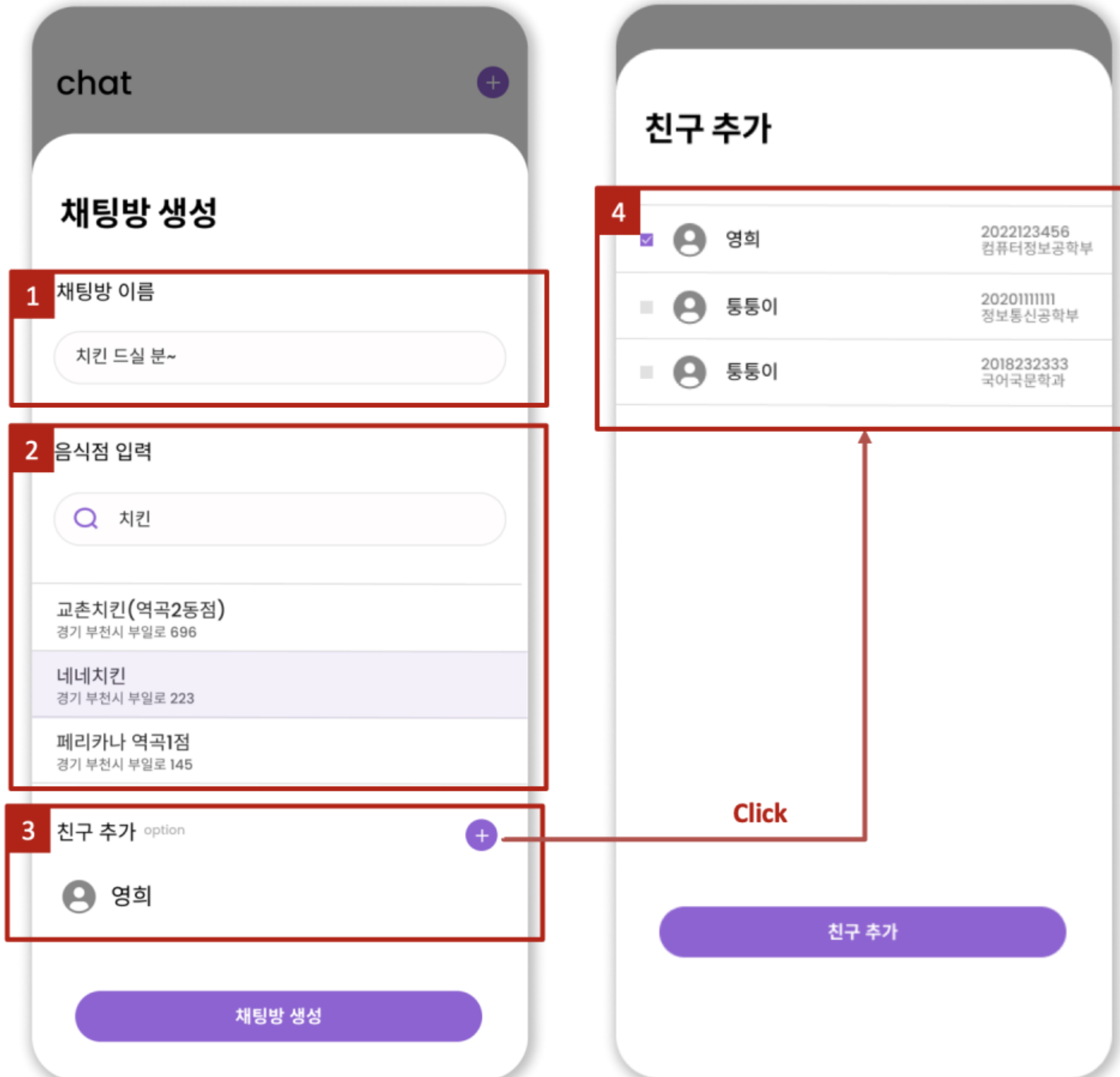
- 음식 이름 및 음식점 이름으로 원하는 음식점을 검색함
- 검색된 역곡 주변 음식점 목록 중 원하는 음식점을 선택

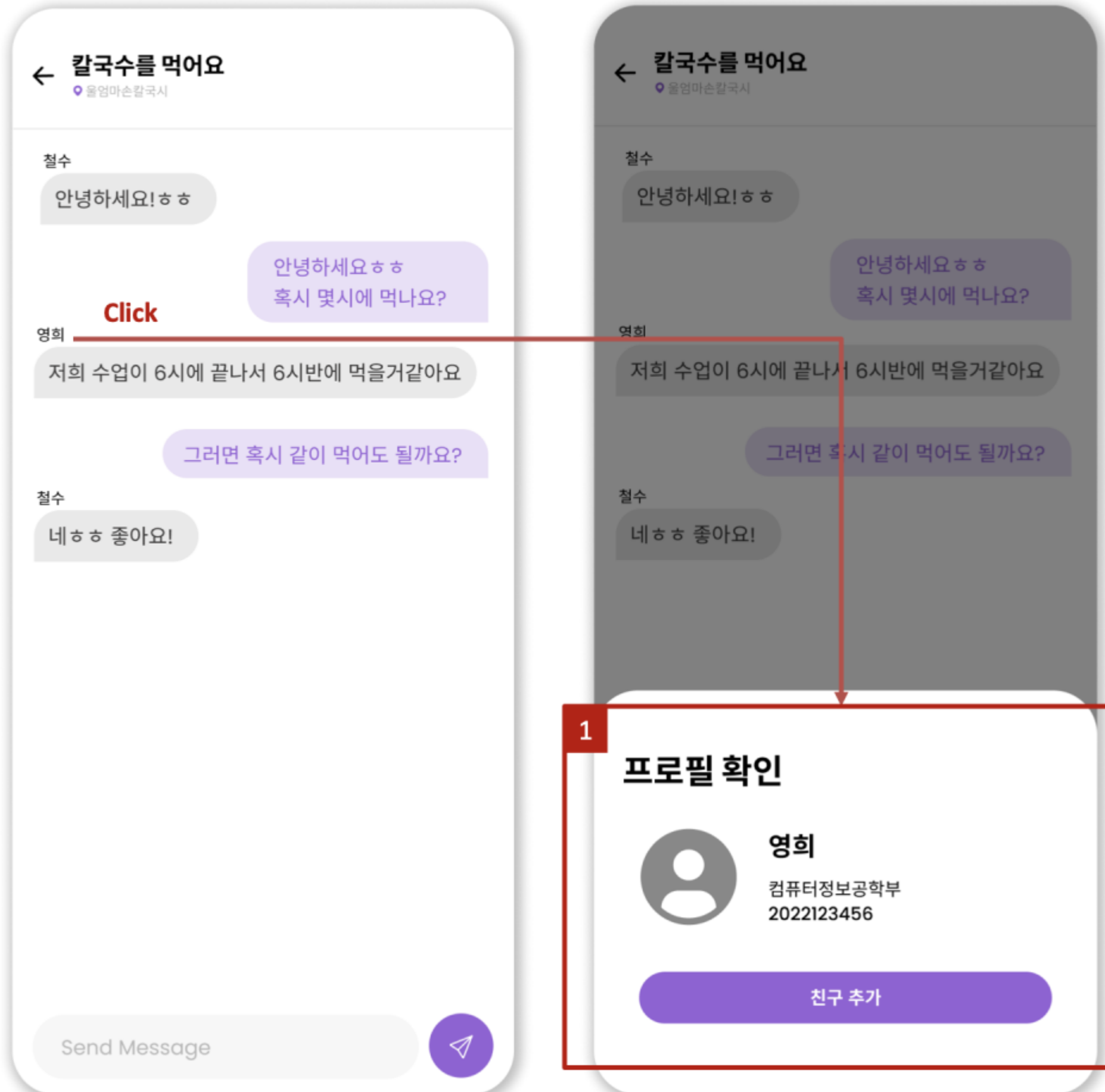
3. 추가된 친구 목록 및 친구 추가 버튼

- 채팅방 생성 시 추가할 친구를 선택함
- 현재 생성할 채팅방에 추가된 친구 목록을 보여줌
- + 버튼을 통해 친구를 추가할 수 있음

4. 친구 추가 페이지

- + 버튼 클릭 시 출력되는 친구를 추가할 수 있는 페이지
- 현재 내 친구 목록의 친구들 중 추가할 친구를 선택할 수 있음





채팅방 화면 및 프로필 확인 - ChatRoomScreen

채팅방 클릭 시 보여지는 채팅 화면
상대 이름 클릭 시 프로필 확인 및 친구 추가 가능

1. 프로필 확인 및 친구 추가

- 채팅 화면에서 상대 이름 클릭 시 프로필을 확인할 수 있음
- 프로필 확인 페이지에서 친구 추가 버튼 클릭 시 친구 추가를 할 수 있음

Sign in

1 id
XXXXXXXXXX

pw
XXXXXXXXXX

로그인

2 회원가입

로그인 화면 - SignInScreen

로그인을 할 수 있는 화면
앱을 처음 시작할 경우 보여지는 화면으로 필요에 따라 회원 가입 화면으로 이동 가능

1. 로그인 정보 입력 및 로그인

- 로그인 정보(id, pw)를 입력하고 로그인 버튼을 클릭하여 로그인 함

2. 회원가입 화면 이동

- 클릭 시 회원 가입 화면으로 이동할 수 있음

Sign up

1 id
XXXXXXXXXX

pw
XXXXXXXXXX

인적사항 입력

이름
김철수

학번
2022123456

전공
컴퓨터정보공학부

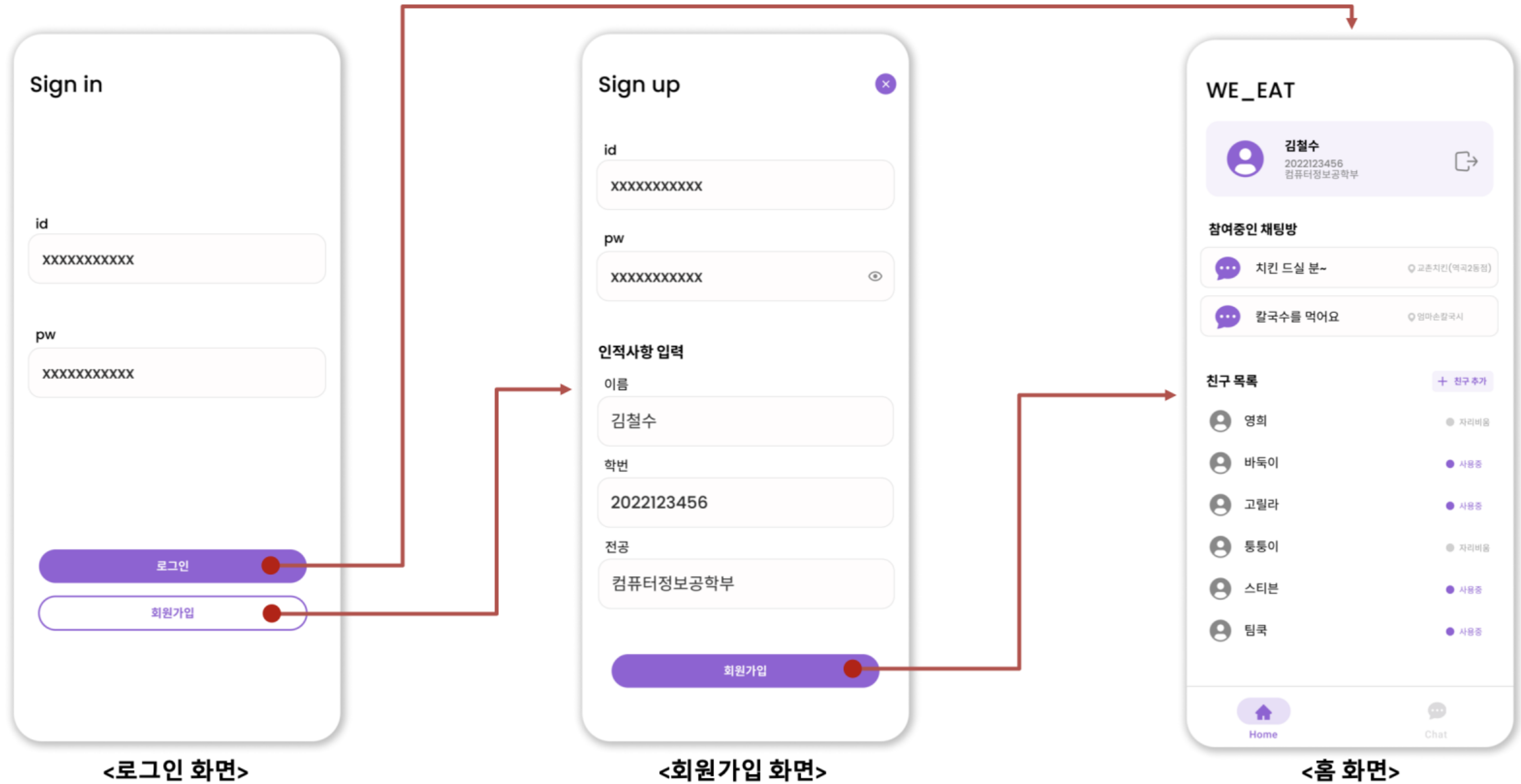
회원가입

회원가입 화면 - SignUpScreen

회원가입을 할 수 있는 화면
필수 입력 사항(id, pw, 이름, 학번, 전공)을 입력한 뒤 회원가입 가능

1. 회원가입 정보 입력 및 회원가입

- 회원가입 시 필요한 정보(id, pw, 이름, 학번, 전공) 입력란
- 모든 정보 기입 후 회원 가입 버튼을 클릭하여 회원가입 진행

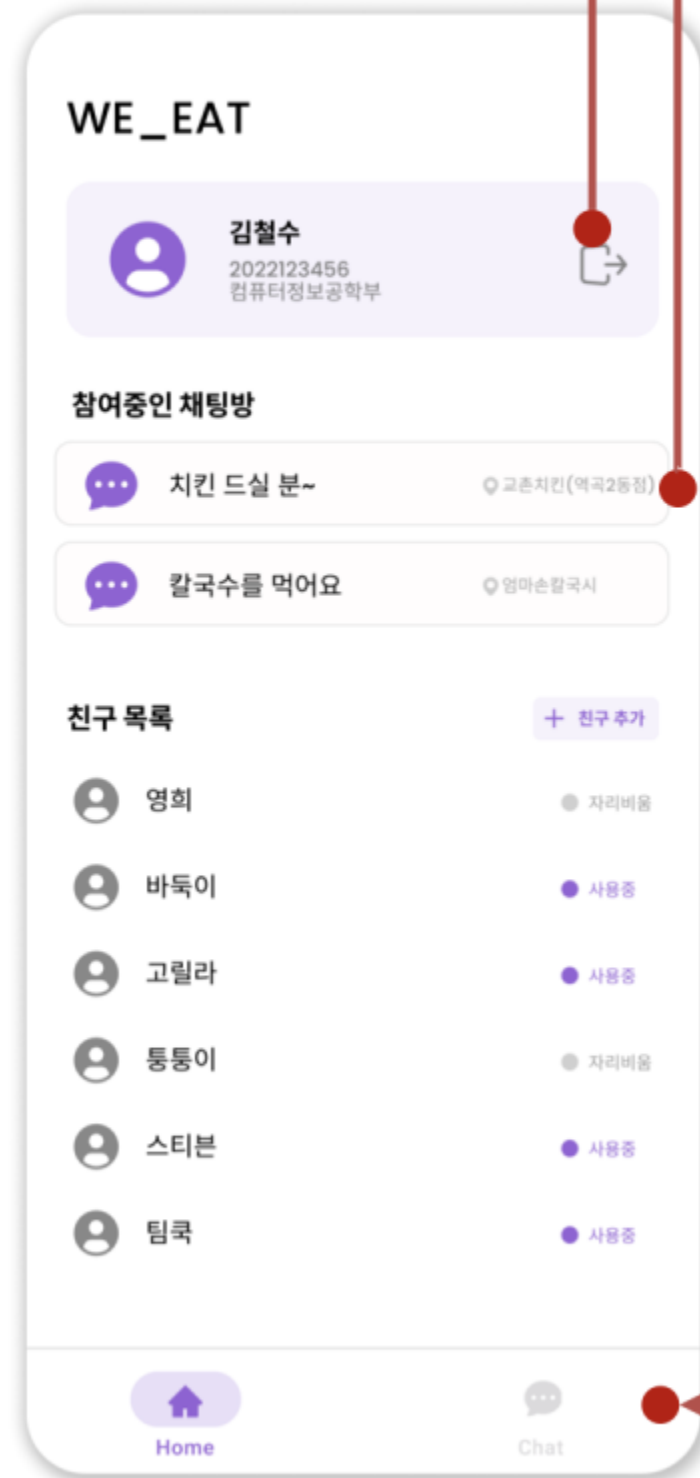


06 GUI 설계

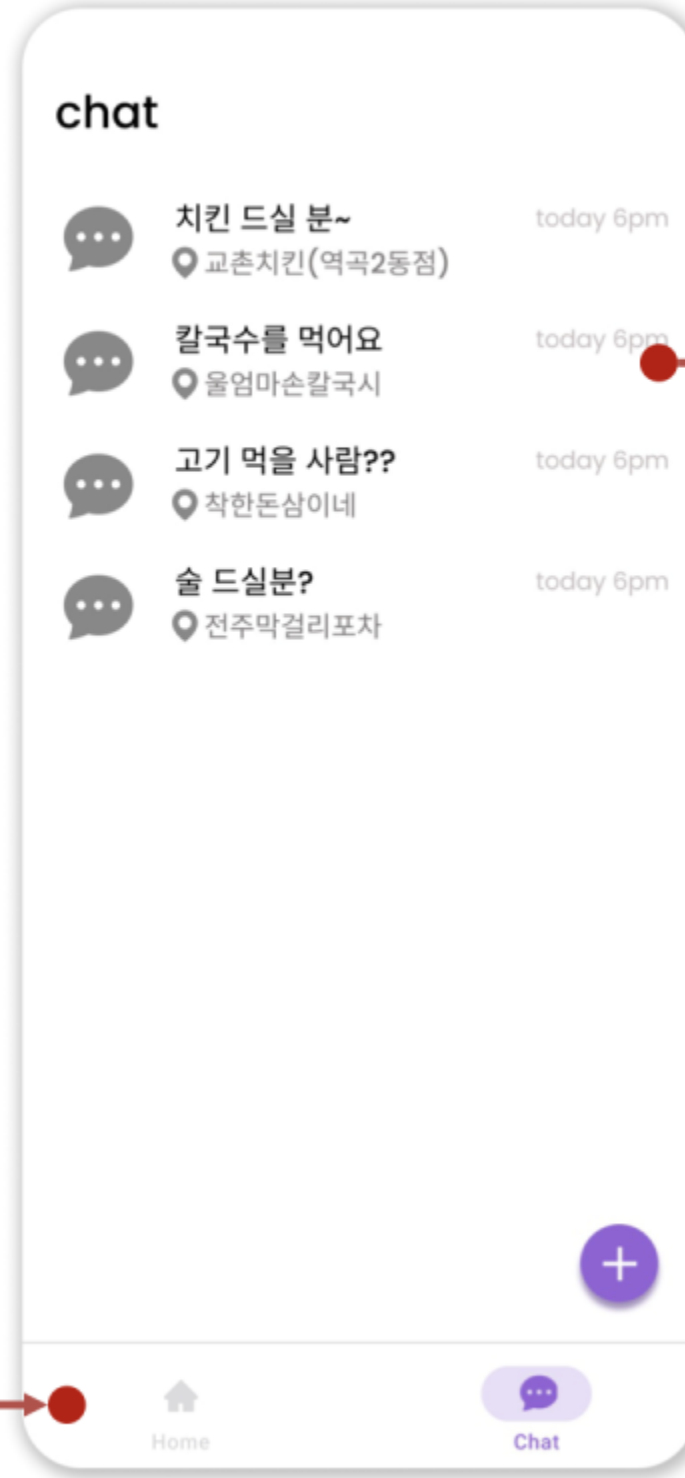
01. UI 설명

02. 흐름 별 UI 이동

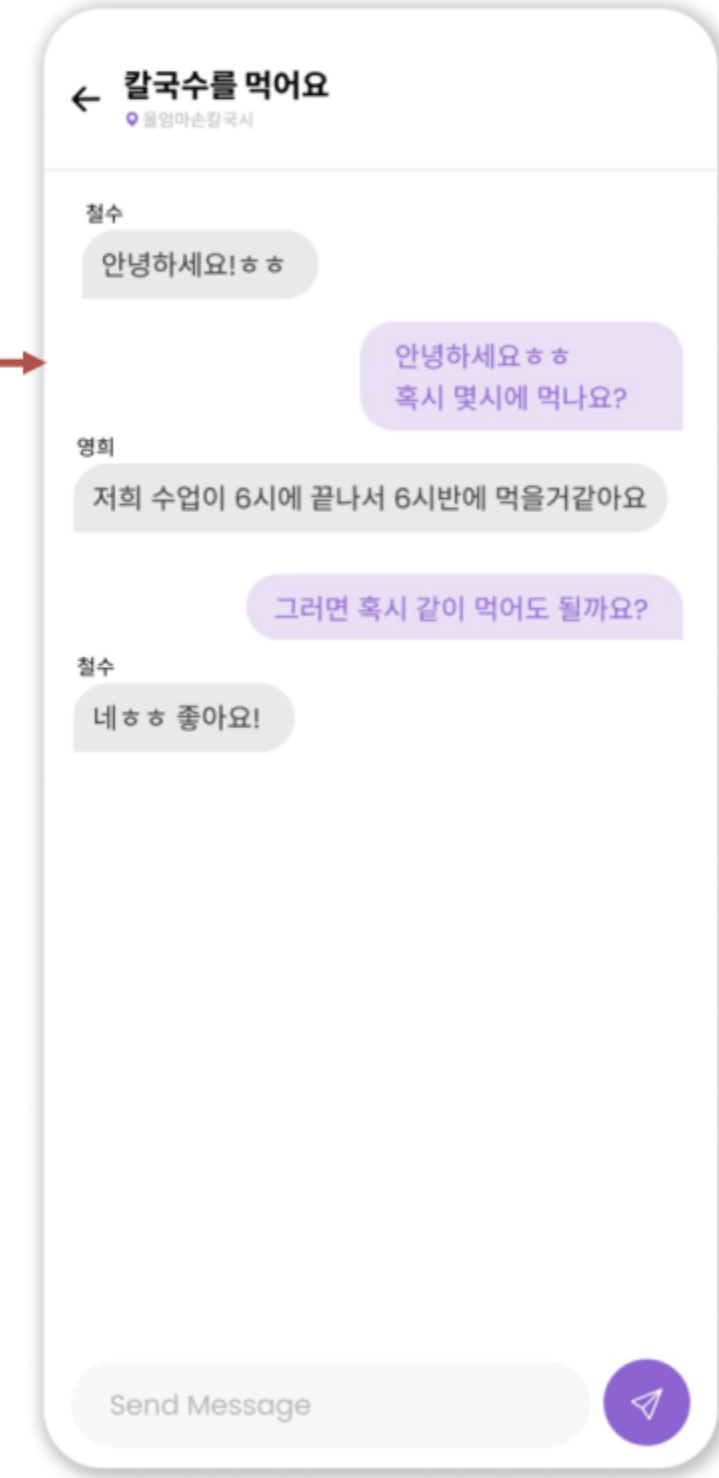
<로그인 화면>



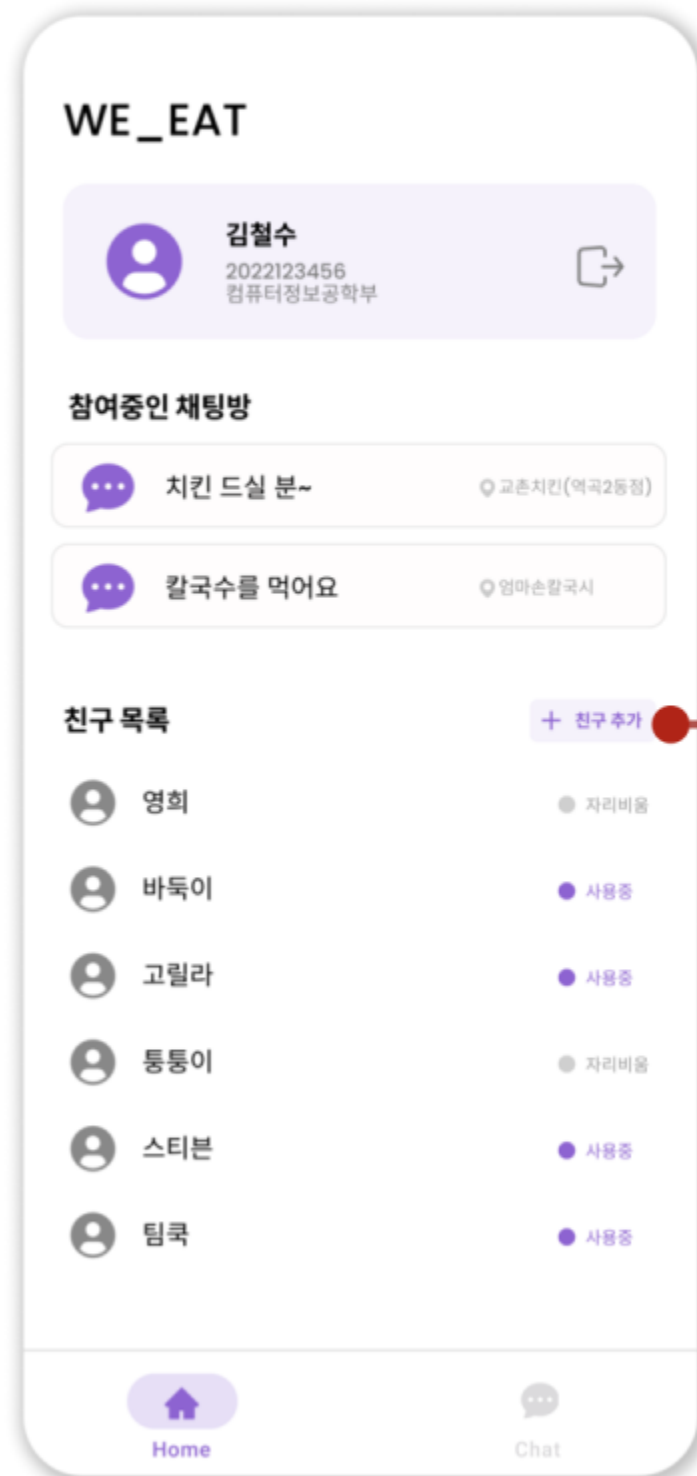
<홈 화면>



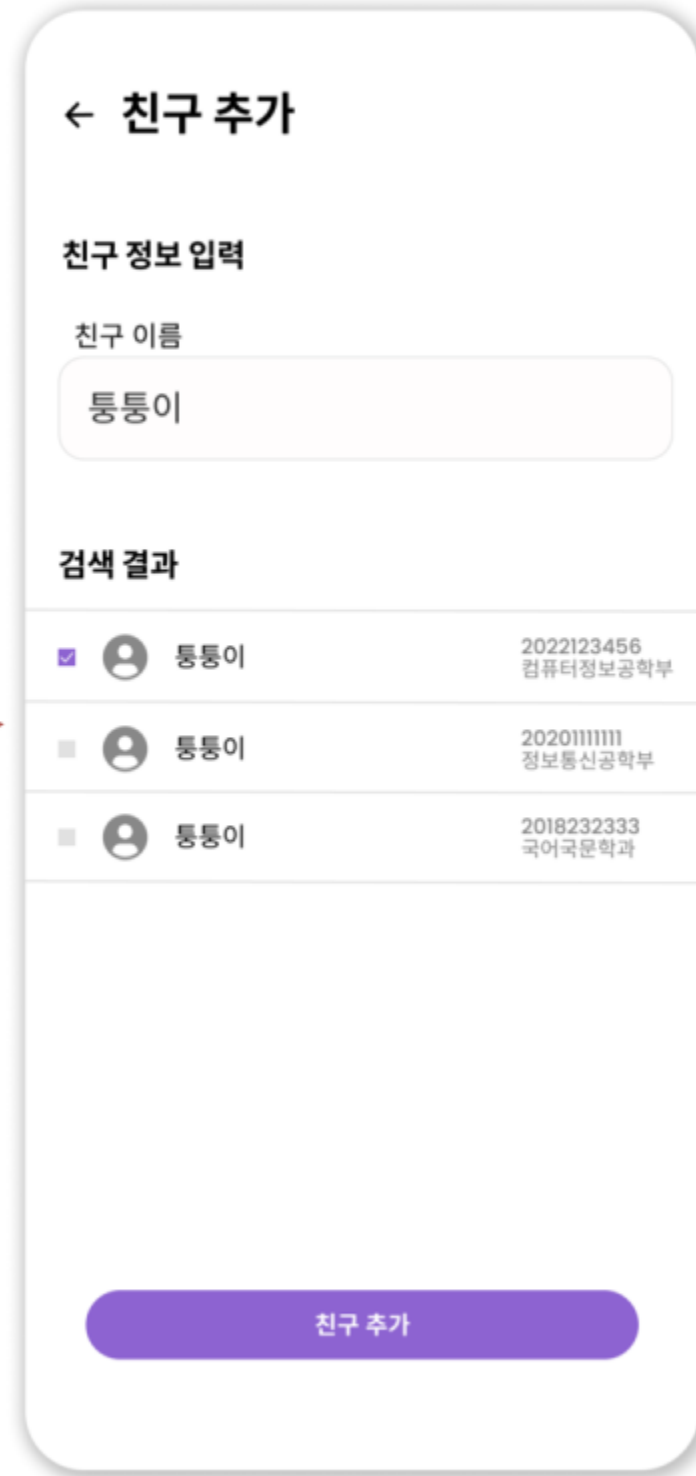
<채팅 목록 화면>



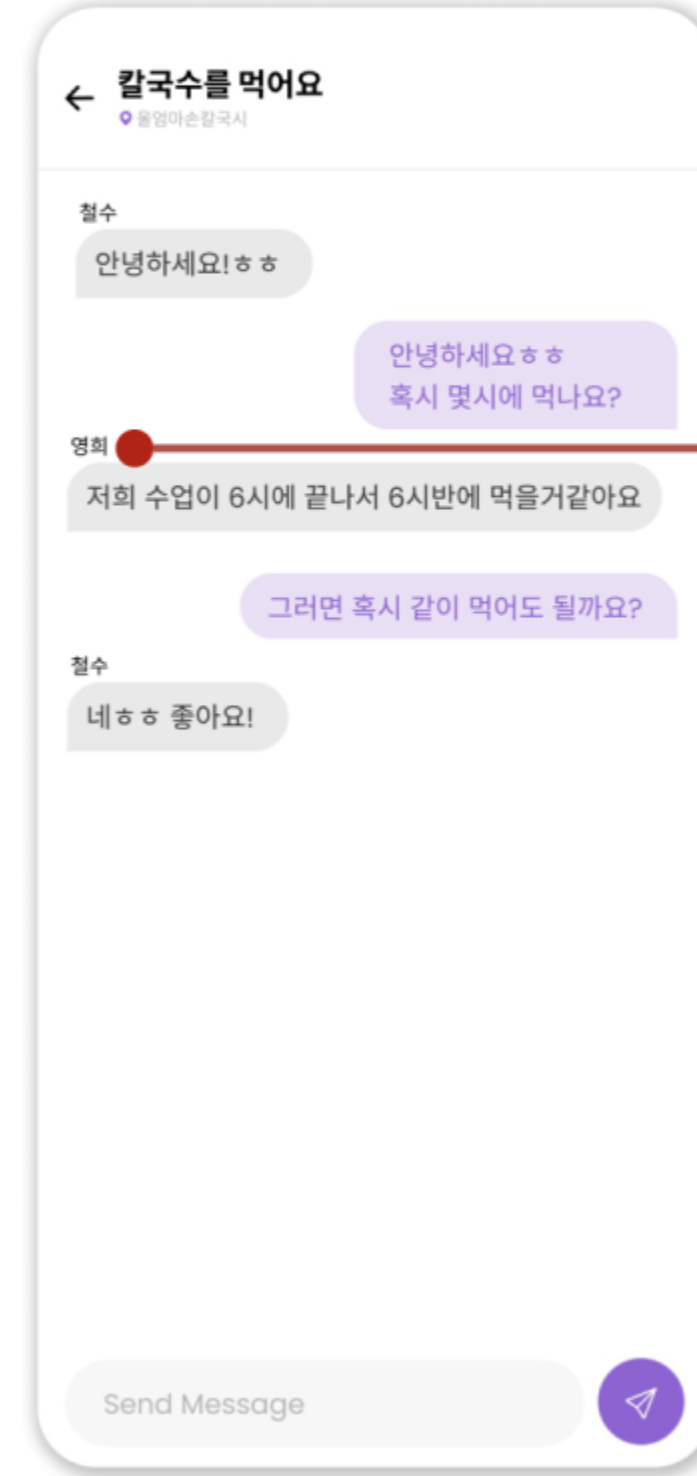
<채팅방 화면>



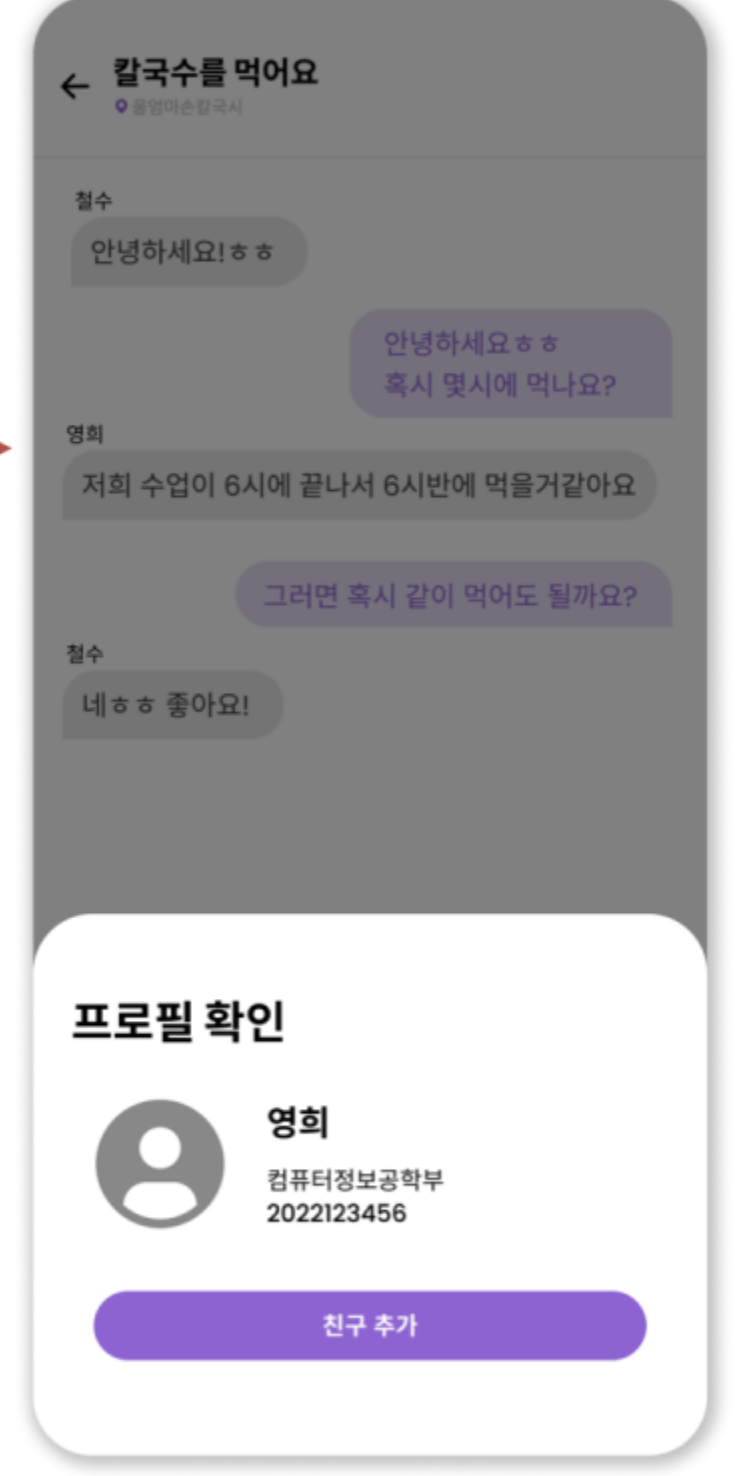
<홈 화면>

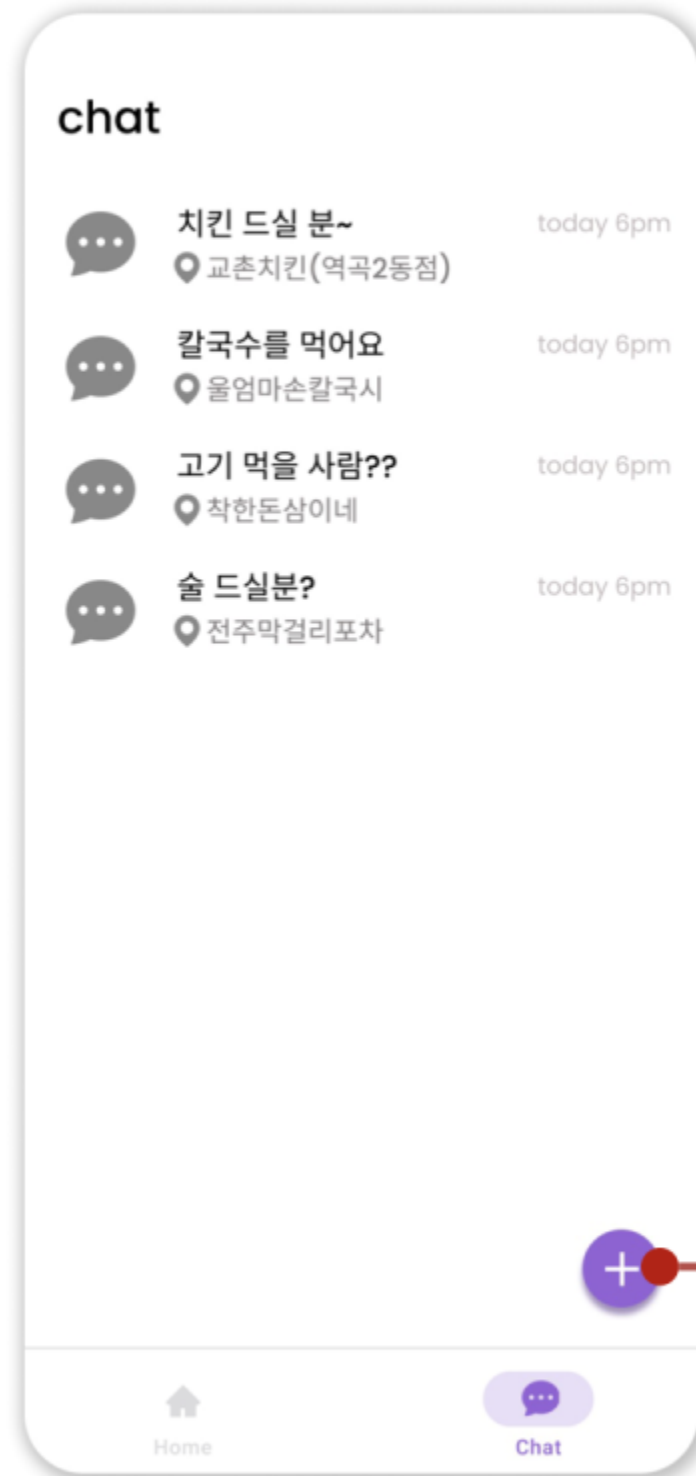


<친구 추가 화면>

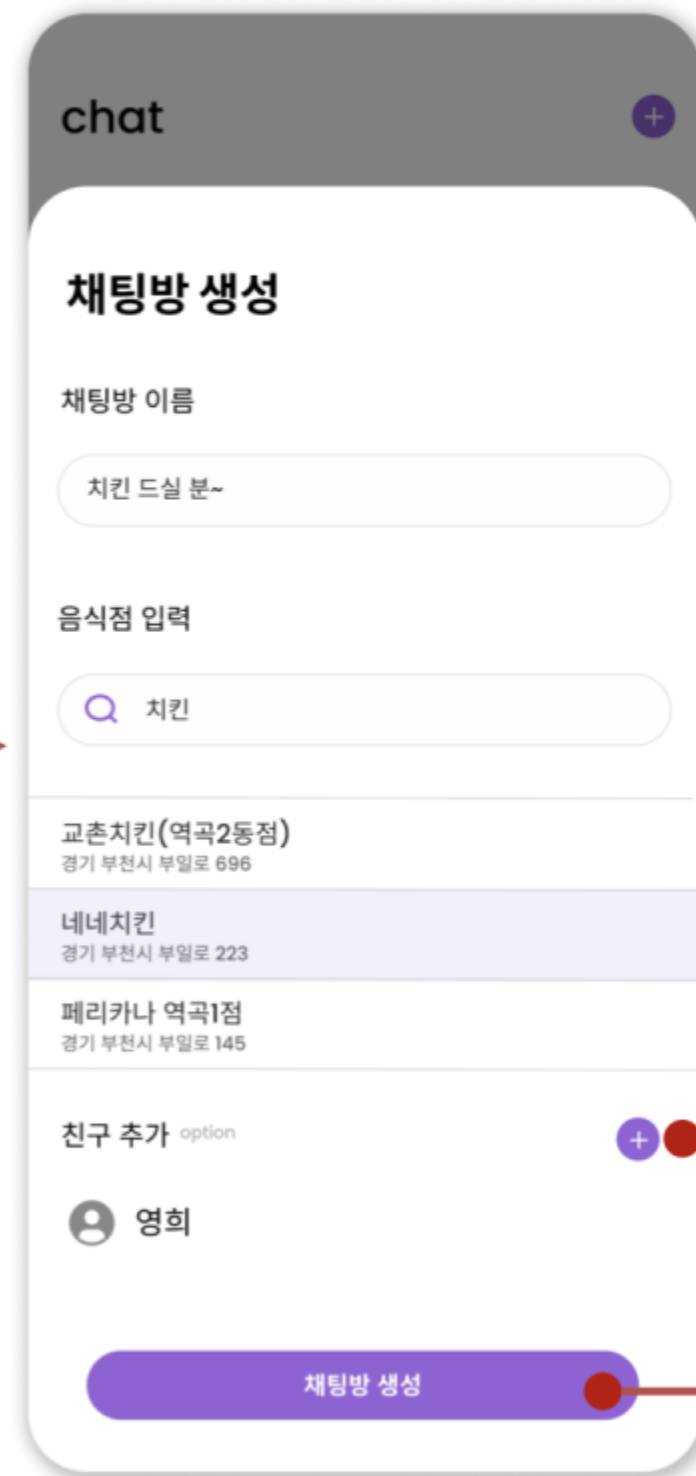


<채팅방 화면 - 프로필 확인>

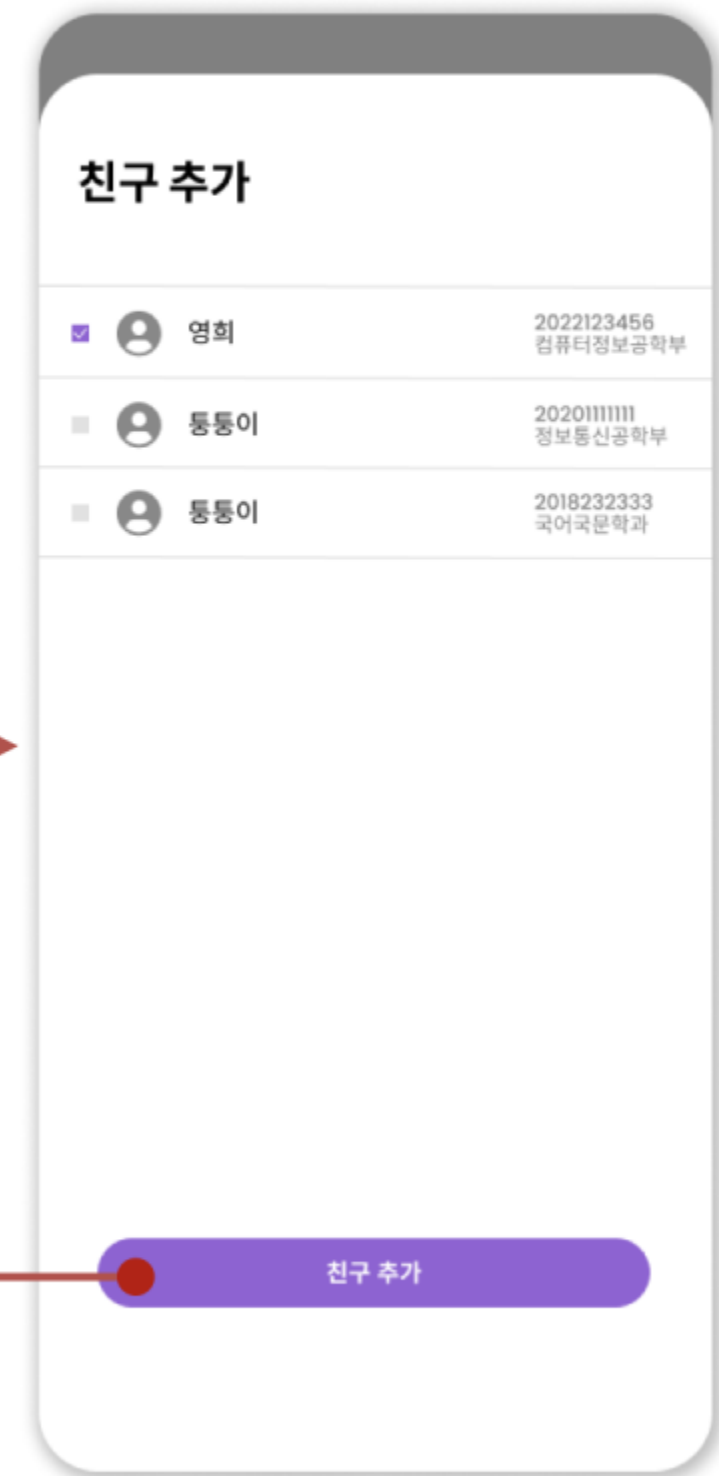




<전체 채팅방 목록 화면>



<채팅방 생성 화면>



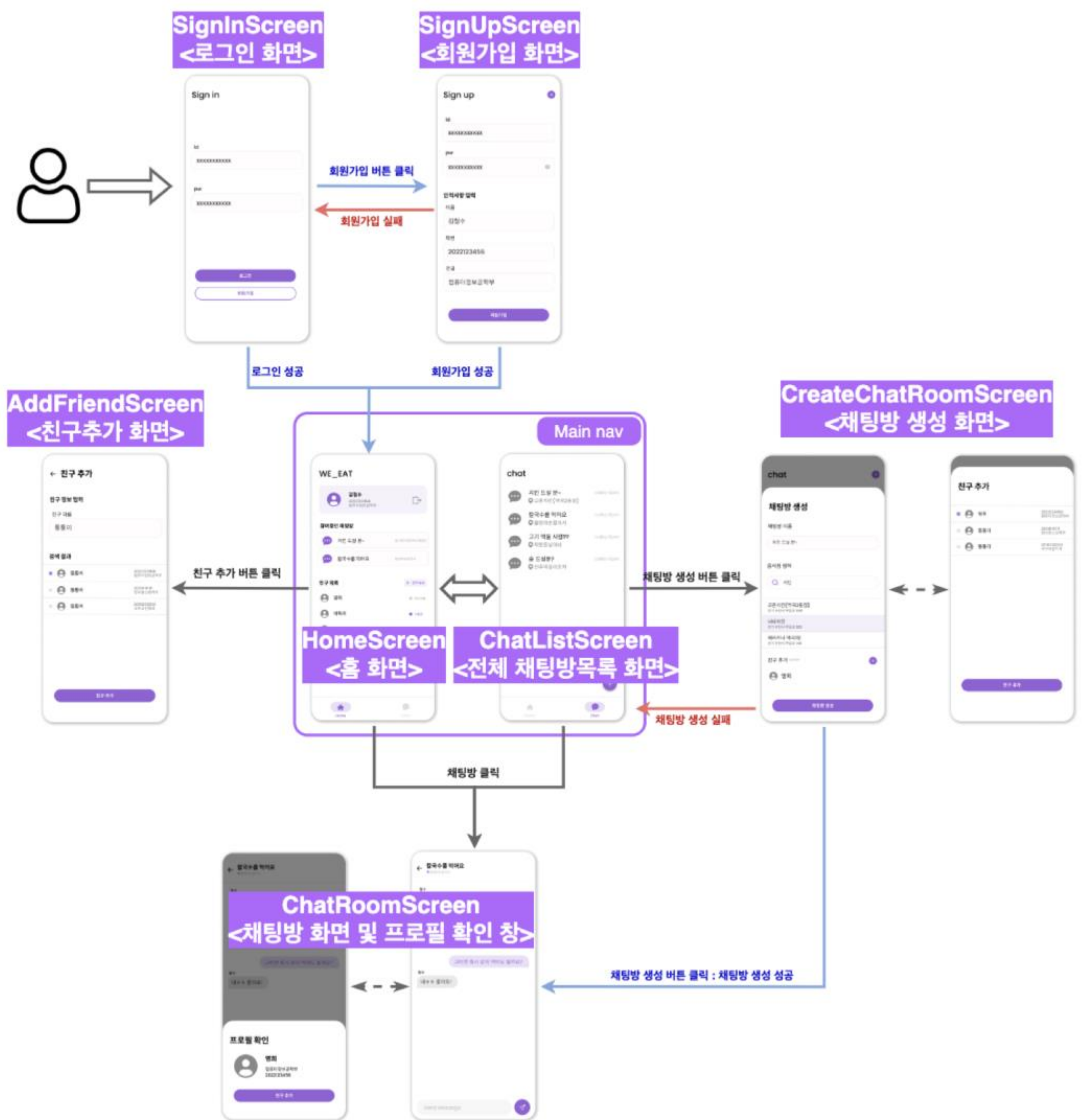
<채팅방 생성 화면 - 친구 추가>

<채팅방 화면>

06 GUI 설계

01. UI 설명

02. 흐름 별 UI 이동



07 requirements vs implementation

01. requirements 02. implementation

requirements	description
회원가입	사용자 정보를 입력하여 회원가입을 합니다.
로그인	회원가입된 사용자가 id, pw를 입력하여 로그인합니다.
로그아웃	로그인 상태의 사용자가 로그아웃 버튼을 클릭하여 로그아웃 합니다.
친구추가	이름을 검색하여 사용자를 찾고 친구를 추가합니다.
친구상태 실시간 확인	친구들의 로그아웃, 로그인 상태를 확인합니다.
채팅방 생성	음식점을 기반으로 채팅방을 생성합니다.
음식점 검색	음식이름을 기반으로 역곡(가톨릭대학교) 주변 음식점을 검색합니다.
채팅방 목록 확인	현재 생성된 채팅방 목록을 확인합니다. / 기존 참여중인 채팅방 목록을 확인합니다.
채팅방 참여	채팅방을 클릭하여 입장합니다.
1:n 채팅	입장한 채팅방에서 웹소켓을 통해 채팅합니다.
프로필 확인	상대방의 프로필을 클릭하여 사용자 정보를 확인합니다.
채팅방 나가기	현재 참여중인 채팅방에서 완전히 나갑니다.

07 requirements vs implementation

01. requirements

02. implementation

requirements	implementation
회원가입	0
로그인	0
로그아웃	0
친구추가	0
친구상태 실시간 확인	0
채팅방 생성	0
음식점 검색	0
채팅방 목록 확인	0
채팅방 참여	0
1:n 채팅	0
프로필 확인	0
채팅방 나가기	0

감사합니다

임옥이공