

# CKIE-APP

**Convergent Capstone Design 2**  
**Week 8 - Progress Report**

**Group 6 Park JongBeum & Baek SeungHeon, October 23rd 2023**

# Recall

# Recall

## What we've done last week

- Location information in the Background
- ChatRoom UI using Flyer
- socket.io chatting specification
  - create\_room
  - invite\_room
  - join\_room
- Mongoose Type Inference
- Minor: Android FCM Test

## 백그라운드 실행 확인

아래 앱과 서비스는 백그라운드에서 실행 중입니다.



cookie\_app

3분

중지

완료



# Recall

## What we've done last week

- Location information in the Background
- ChatRoom UI using Flyer
- socket.io chatting specification
  - create\_room
  - invite\_room
  - join\_room
- Mongoose Type Inference
- Minor: Android FCM Test

9:49

채팅방 추가

채팅방 이름  
asdfasdf

친구 3명

- test1
- test1
- test1

**asdfasdf (4명)**

채팅방을 개설하겠습니까?

취소 확인

8:38

채팅


20:36

Test Message Here

Hello World

[mongoosejs.com](https://mongoosejs.com)

Mongoose ODM v7.5.2



Message

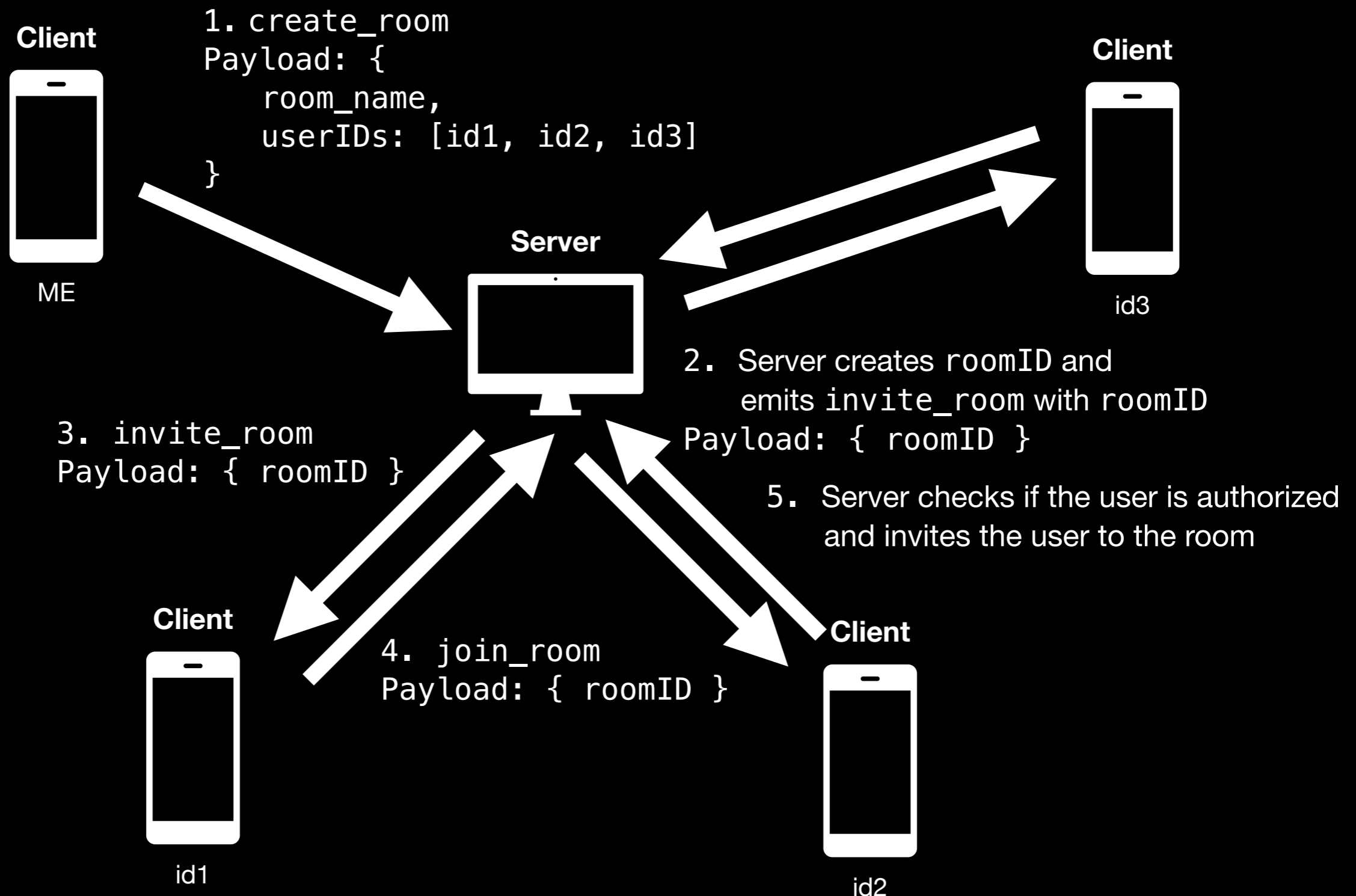
# Recall

## What we've done last week

- Location information in the Background
- ChatRoom UI using Flyer
- socket.io chatting specification
  - create\_room
  - invite\_room
  - join\_room
- Mongoose Type Inference
- Minor: Android FCM Test

# Creating Room

## create\_room process





# Recall

## What we've done last week

- Location information in the Background
- ChatRoom UI using Flyer
- socket.io chatting specification
  - create\_room
  - invite\_room
  - join\_room
- Mongoose Type Inference
- Minor: Android FCM Test

# Progress

# Maps Tab

# Maps Tab

## Need to share location information with friends

- We will display the location information of our friends with a "marker"
- Location information should be known immediately
- So we use "socket.io"

# Maps Tab

## socket.io in Flutter



```
1 class MapInfoRequest {
2   double latitude;
3   double longitude;
4
5   MapInfoRequest({
6     required this.latitude,
7     required this.longitude,
8   });
```



```
1 class MapInfoResponse {
2   String userid;
3   double latitude;
4   double longitude;
5
6   MapInfoResponse({
7     required this.userid,
8     required this.latitude,
9     required this.longitude,
10  });
```

# Maps Tab

socket.io in Flutter



```
1 class MapEvents {  
2   static const position = 'position';  
3 }
```

# Maps Tab

## socket.io in Flutter



```
1 void position() {  
2   socket.emit(  
3     MapEvents.position,  
4     MapInfoRequest(  
5       latitude: _currentLocation.latitude,  
6       longitude: _currentLocation.longitude,  
7     ).toJson(),  
8   );  
9 }
```

# Maps Tab

## socket.io in Flutter



```
1 void _onPosition(data) {  
2     final MapInfoResponse info = MapInfoResponse.fromJson(data);  
3 }
```



# Maps Tab

## socket.io in Server



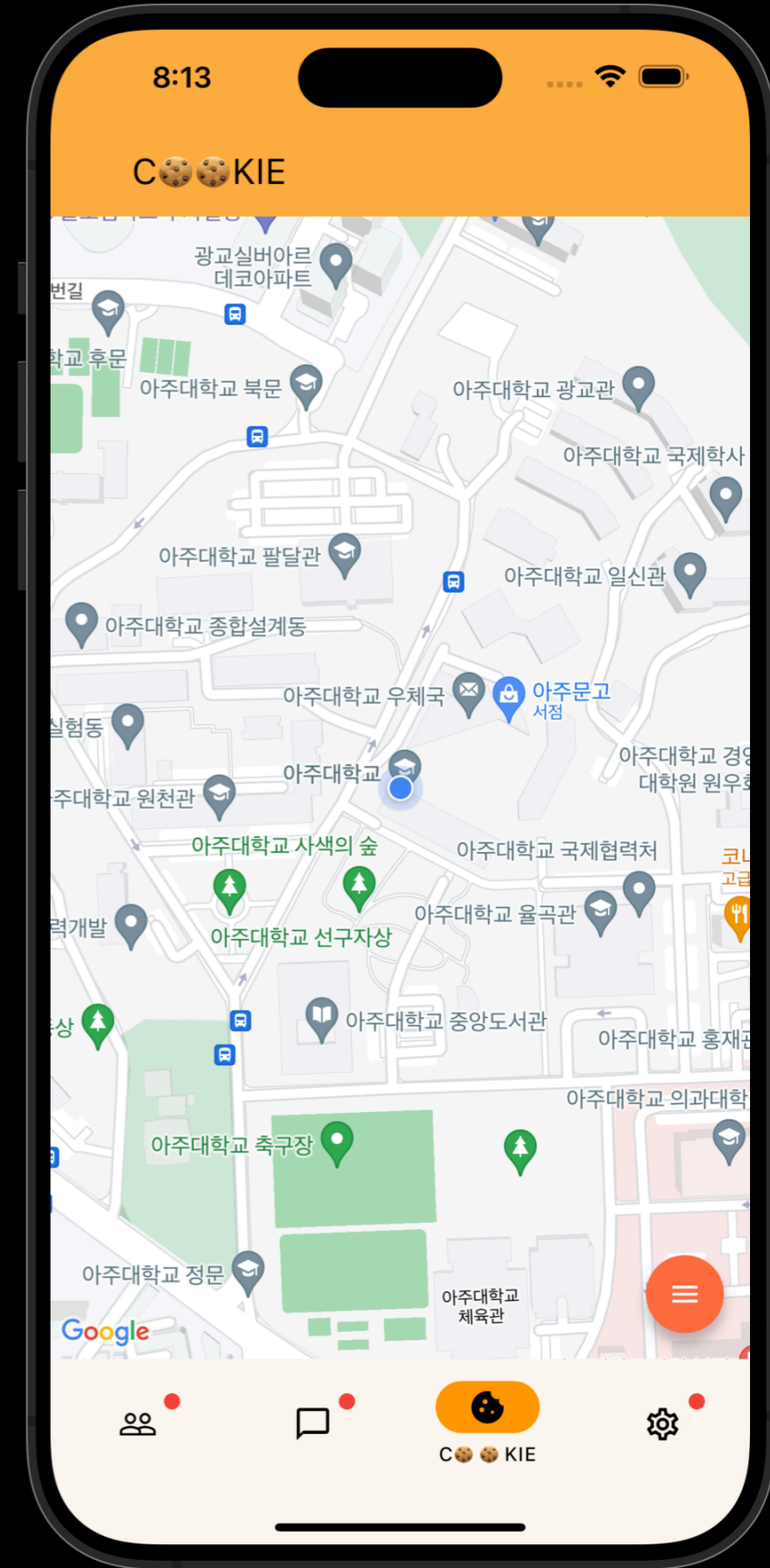
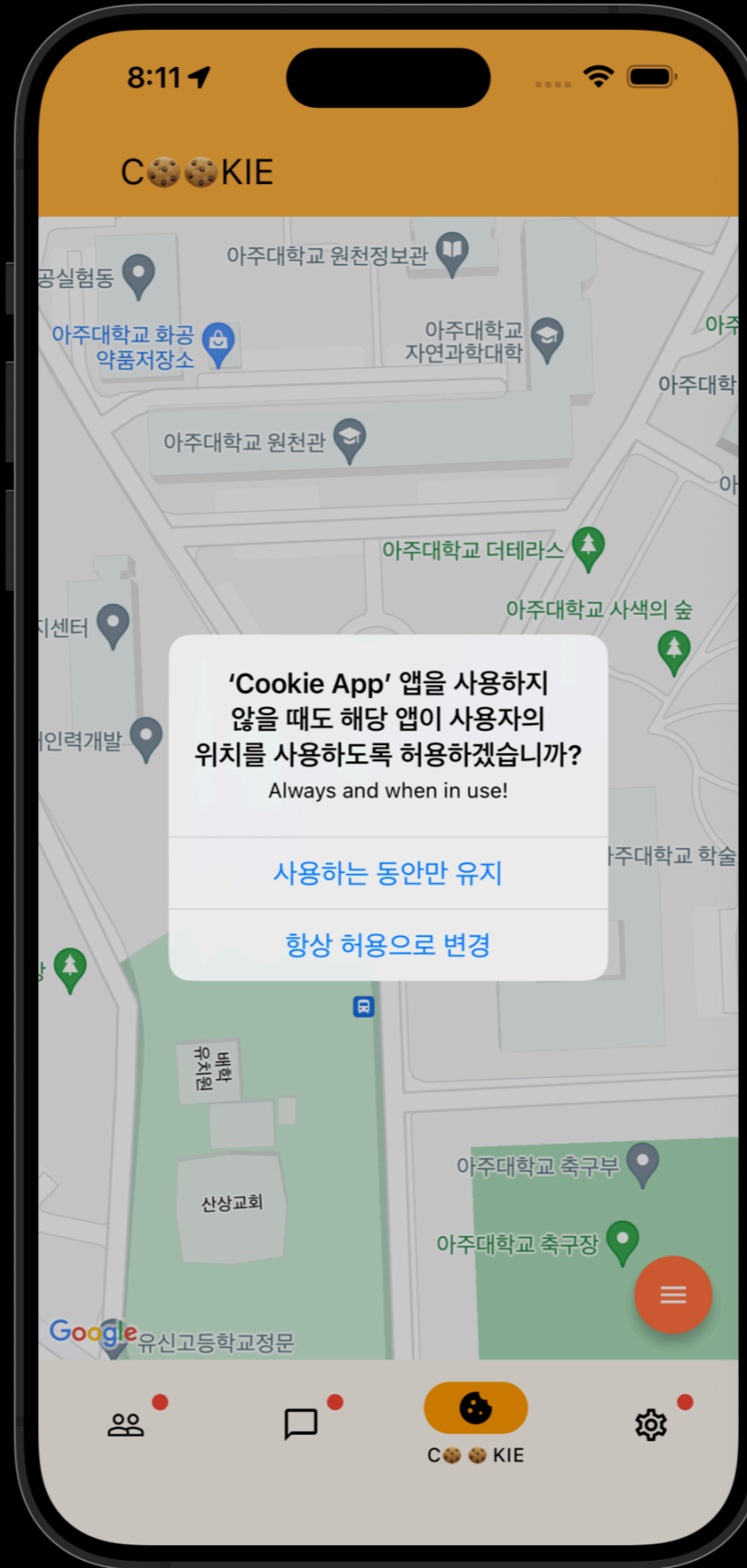
```
1 socket.on(  
2   MapType.MapEvents.position,  
3   async ({ latitude, longitude }: MapType.MapRequest) => {
```

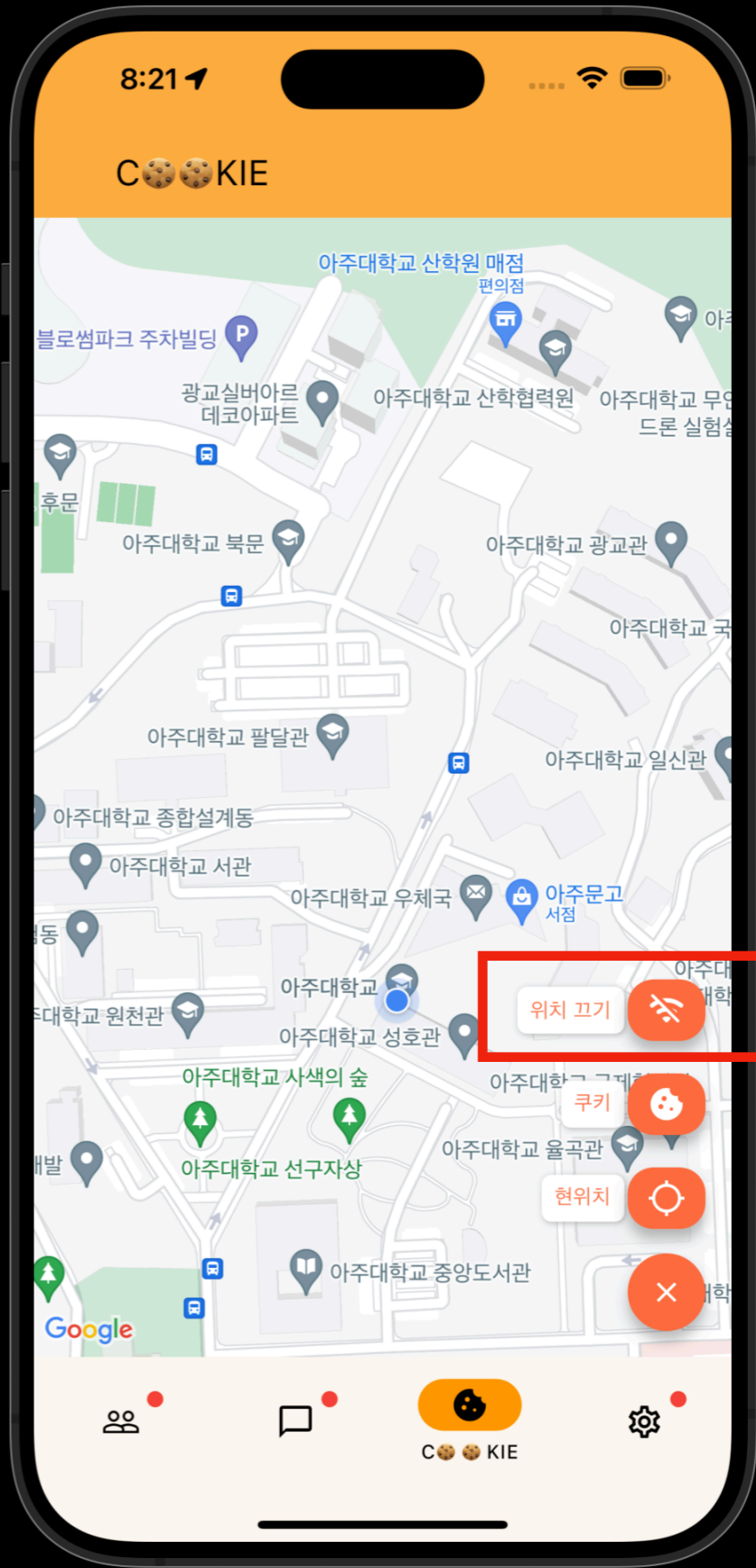
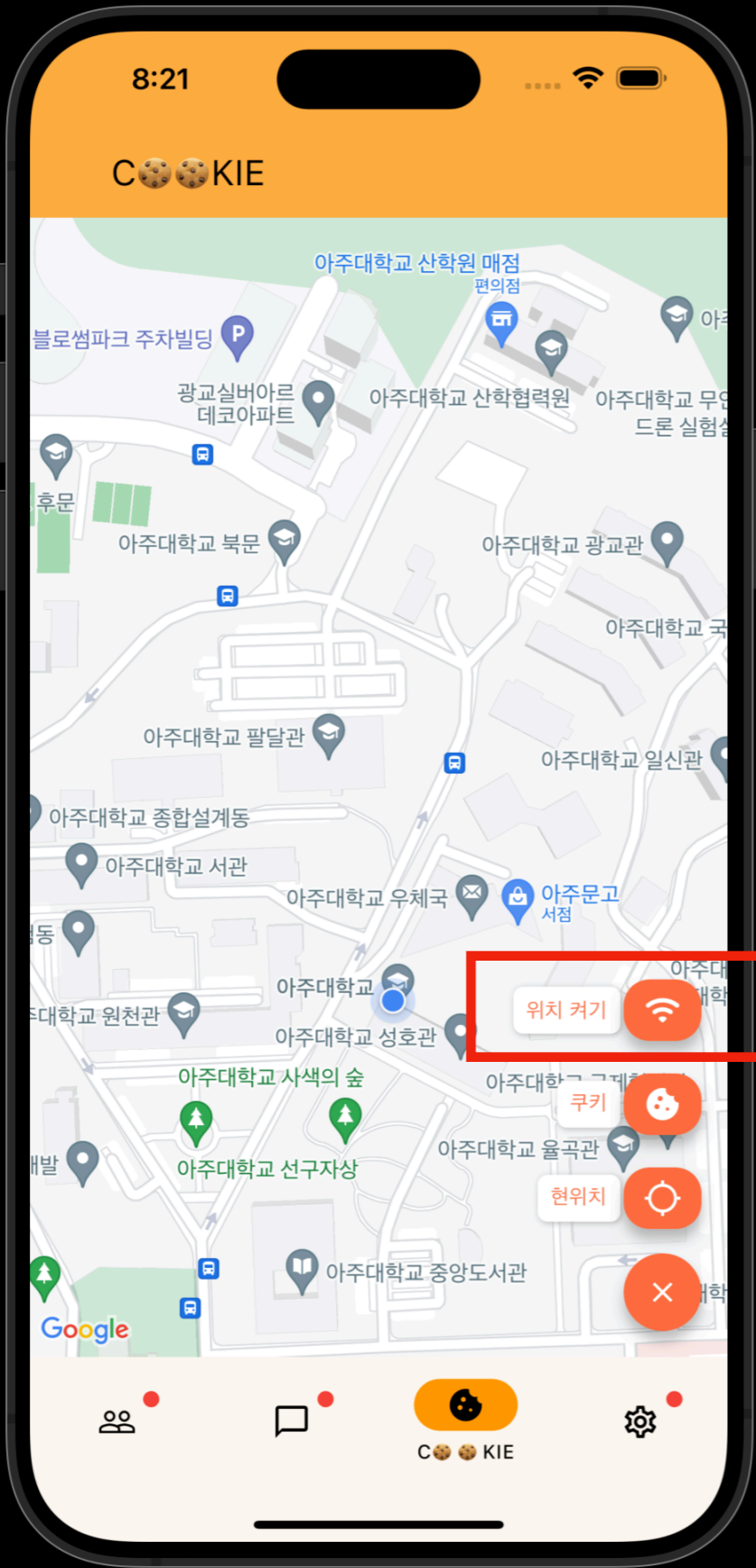
# Maps Tab

## socket.io in Server



```
1  friendList.forEach((friendID) => {  
2    socket.to(friendID).emit(MapEvents.position, {  
3      userid: socket.data.userID,  
4      latitude: latitude,  
5      longitude: longitude,  
6    } as MapType.MapResponse);  
7  });  
8
```






```
testid2 position
location 37.2831 127.0448
sending location to testid1
sending location to testid3
sending location to testid4
testid2 position
location 37.2831 127.0448
sending location to testid1
sending location to testid3
sending location to testid4
```

```
flutter: INFO: 2023-10-22 20:16:53.144366: socket connected
```

```
flutter: INFO: 2023-10-22 20:16:53.156927: socket connected
```

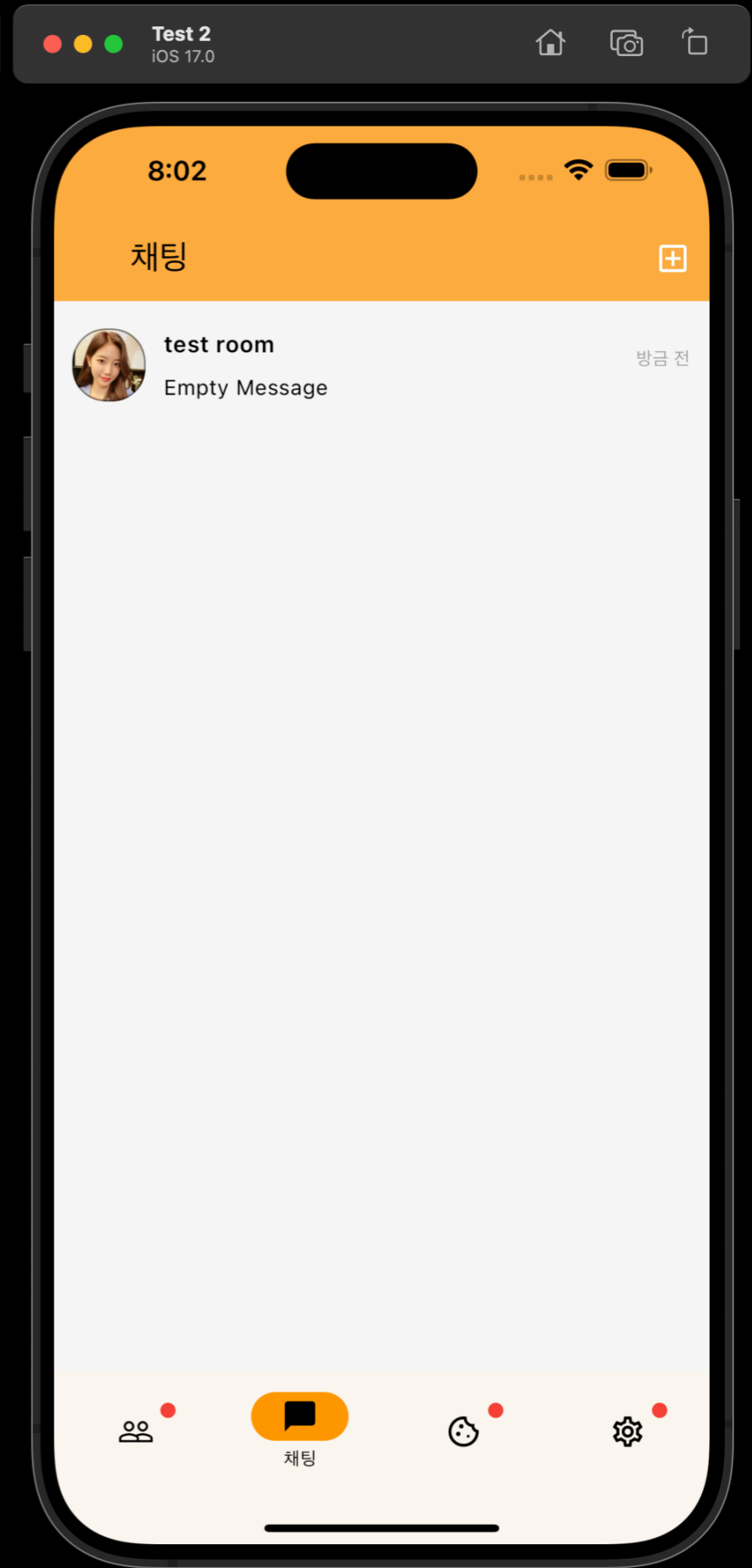
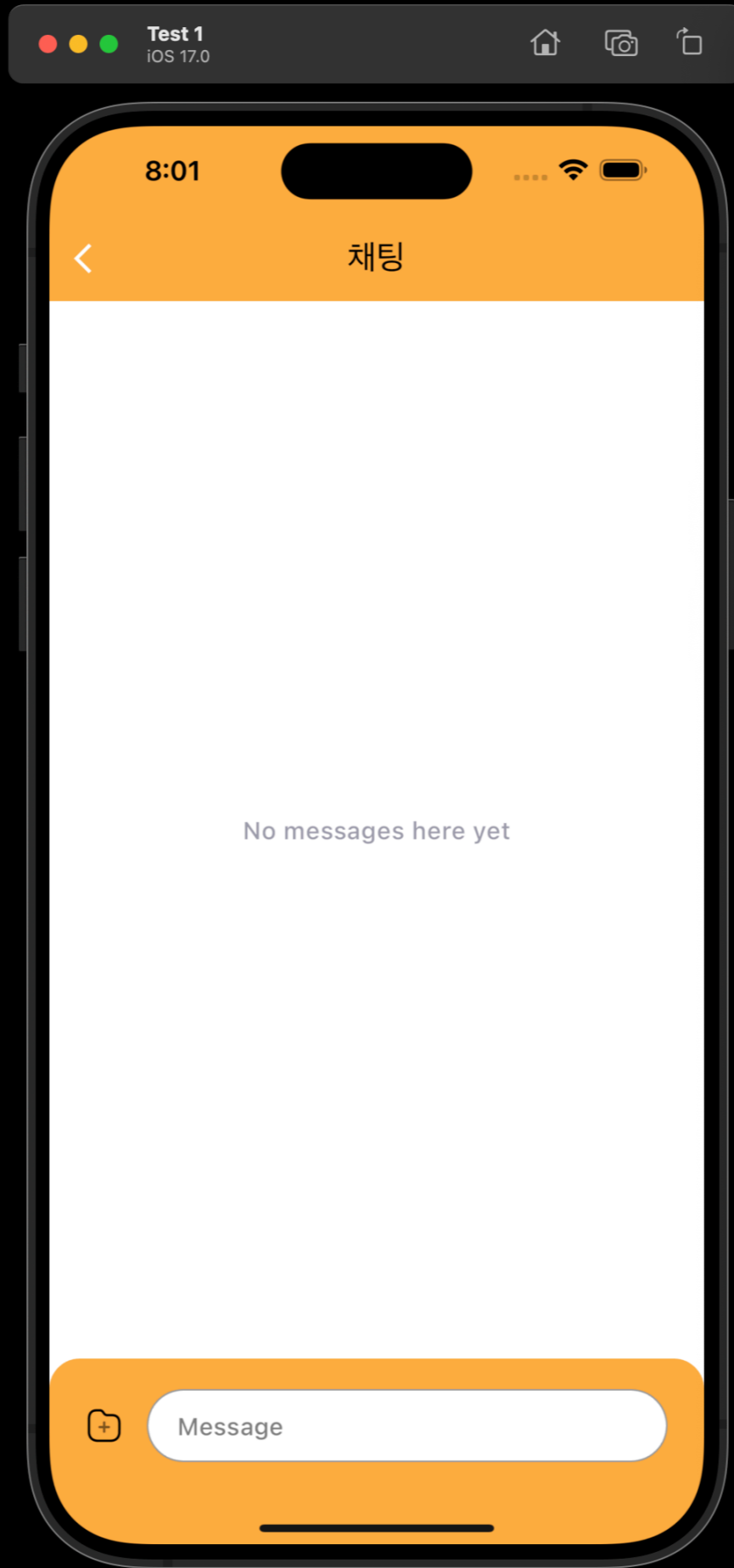
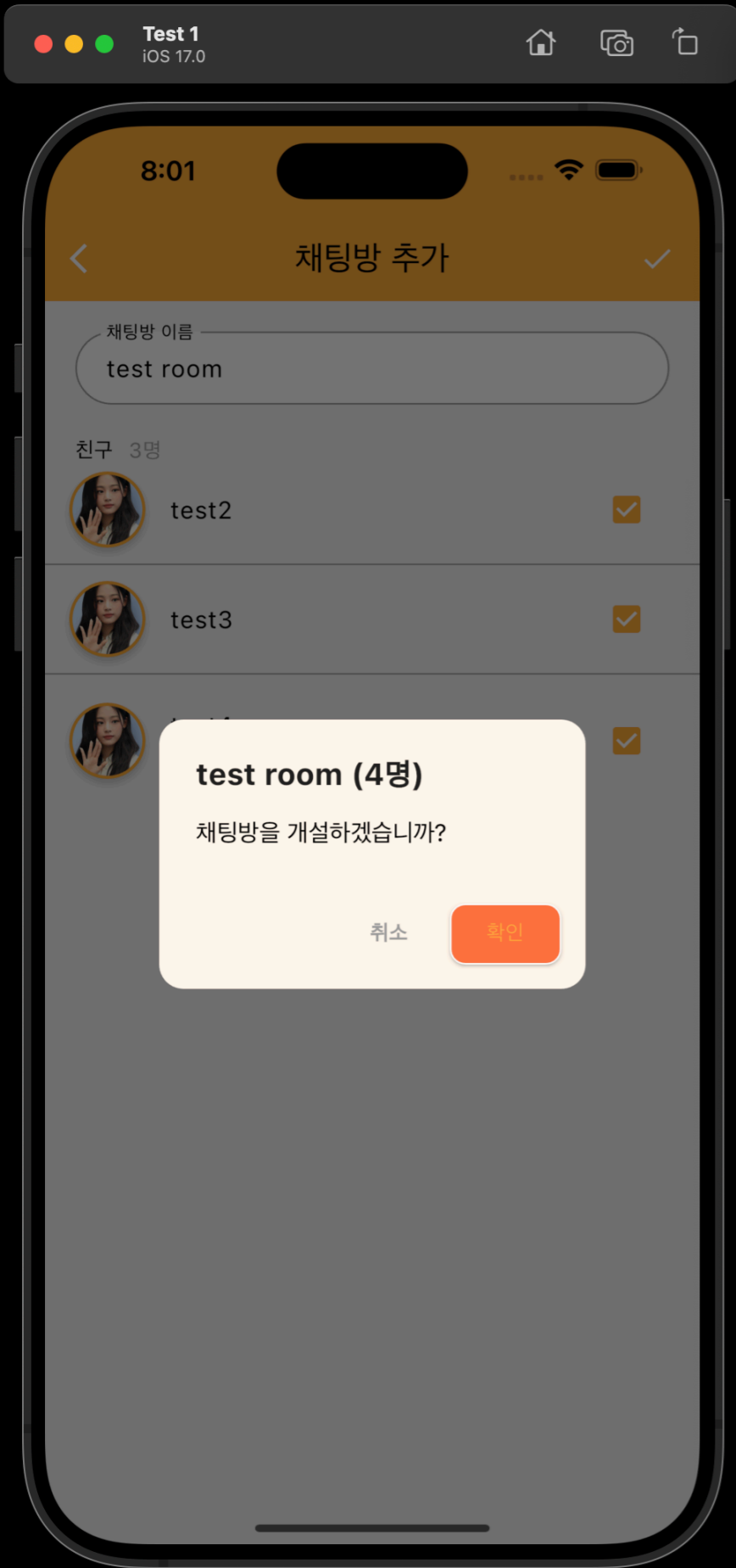
```
flutter: marker_2?: []
```

```
flutter: INFO: 2023-10-22 20:17:09.416333: position: {userid: testid2, latitude: 37.2831, longitude: 127.0448}
```



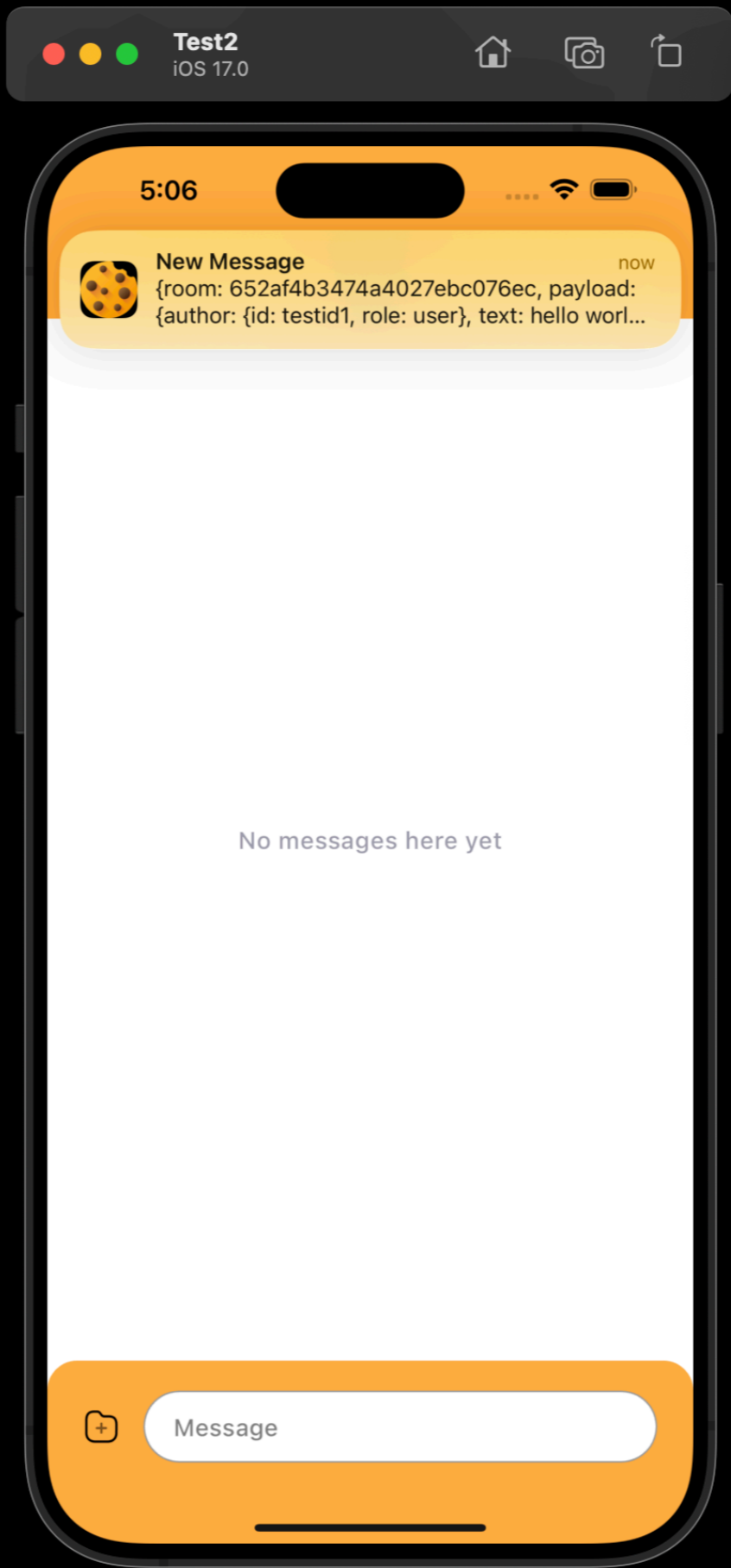
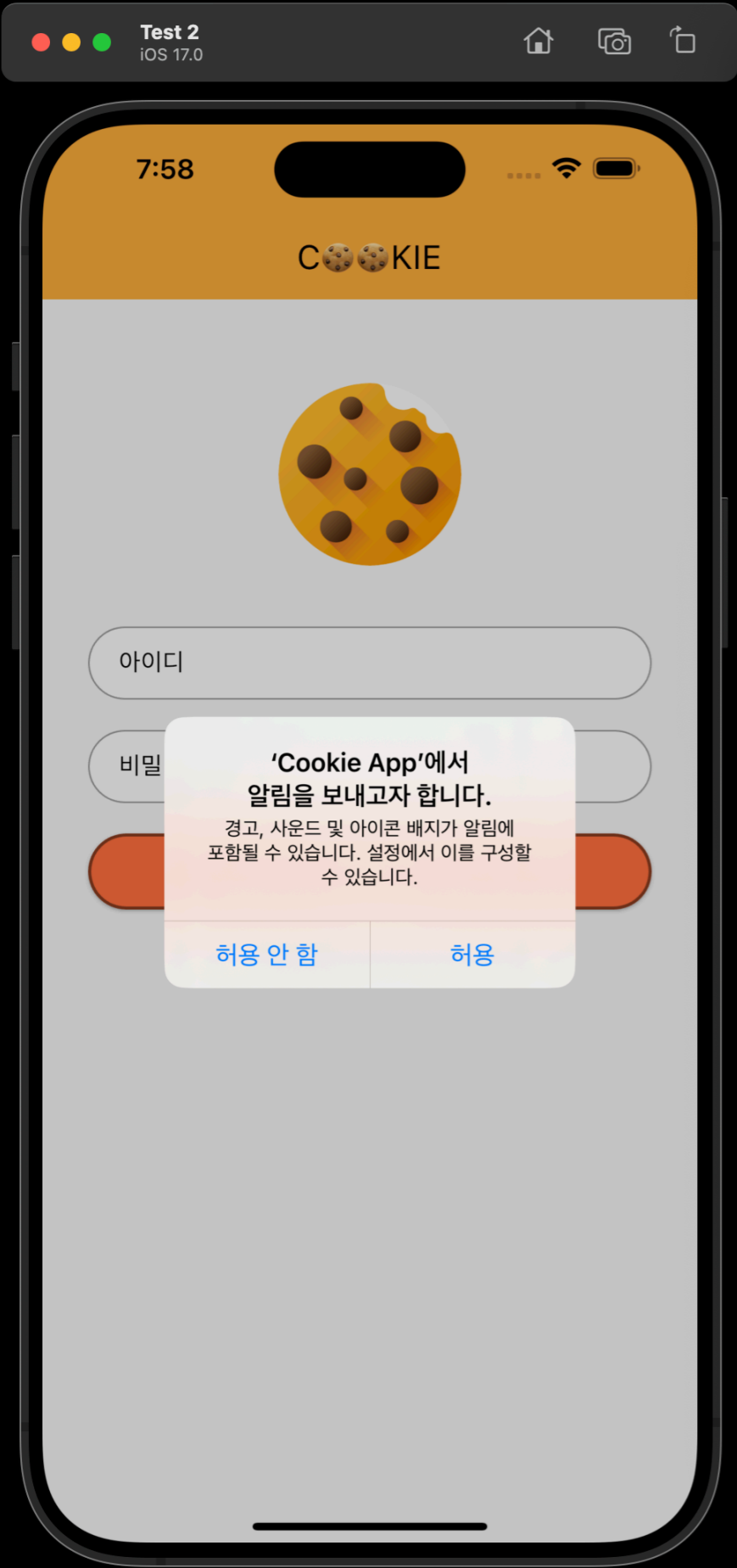
```
position: {userid: testid2,  
latitude: 37.2831, longitude:  
127.0448}
```

# Implementation of Client Chatroom Creation & Join





# iPhone Notification Test



# Flutter Widget Tree - Scope of the Provider

# Bug that we faced

## Scope of the Provider

- Provider of the Authentication Part & Main Service Part should be distinct.
- Even though the user logged out, data are not deleted properly.
- The Problem was on the “Scope of the Provider that should be Responsible of”

# Before

## Just Nesting EVERYTHING

- ThemeProvider
- MapProvider
- AccountViewModel
- PrivateAccountViewModel
- AuthViewModel
- FriendsViewModel
- ChatViewModel
- **Cookie MaterialApp**
  - **AuthPage**
  - **MainPage**

# After

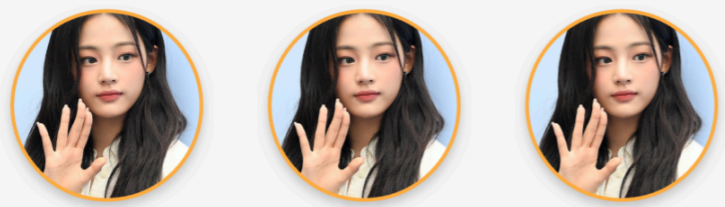
## Nesting Providers Depending on Page

- AuthService
- ThemeProvider
  - **MaterialApp - AuthPage**
  - AccountViewModel
  - PrivateAccountViewModel
  - ChatViewModel
  - FriendsViewModel
  - MapProvider
    - **MaterialApp - MainPage**

# MUI3 Adoption

8:52

친구




test2      test3      test4

친구      2      3      4

7:55

친구



test2      test3      test4

친구      2      3      4



# Thank You

Questions are welcome