

# MeetMate

-Software Design Specification (SDS)-



| Student No. | Name | E-mail             |
|-------------|------|--------------------|
| 22113649    | 김수현  | tndu1212@naver.com |
| 21720870    | 정동윤  | ooehd78@ynu.ac.kr  |
| 21911446    | 정명기  | jmk101711@yu.ac.kr |
| 22112070    | 정은지  | dmdl0518@gmail.com |
| 22113639    | 황윤주  | edelbte@yu.ac.kr   |

## [ Revision history ]

| Revision date | Version # | Description  | Author                |
|---------------|-----------|--|-----------------------|
| 10/28/2023    | 1.0       | First Documentation  | 김수현, 정은지              |
| 11/15/2023    | 1.0.1     | Use case diagram modification  | 김수현                   |
| 11/18/2023    | 1.0.2     | Use case diagram modification,<br>Create additional use cases,<br>Use case description function modification | 김수현                   |
| 11/29/2023    | 1.0.3     | Class Diagram modification,<br>State machine diagram modification  | 정은지, 황윤주              |
| 12/02/2023    | 1.1       | Change logo image<br>State machine diagram modification,<br>Sequence Diagram modification                    | 김수현, 정은지              |
| 12/03/2023    | 1.1.1     | Class Diagram modification,<br>Sequence Diagram modification   | 김수현, 정은지,<br>정명기, 황윤주 |
| 12/04/2023    | 1.1.2     | Sequence Diagram modification  | 김수현                   |
| 12/07/2023    | 1.2       | Final review before deployment<br>Check accuracy   | 황윤주                   |

## = Contents =

|  |     |
|--|-----|
| 1. Introduction .....                  | 1   |
| 2. Use case analysis .....             | 2   |
| 3. Class diagram .....                 | 32  |
| 3.1 Class diagram on Server Side ..... | 32  |
| 3.2 Class diagram on Client Side ..... | 51  |
| 4. Sequence diagram .....              | 71  |
| 4.1 User sequence diagram .....        | 72  |
| 4.2 Post sequence diagram .....        | 77  |
| 4.3 Comment sequence diagram .....     | 83  |
| 4.4 Search Post sequence diagram ..... | 86  |
| 4.5 Invitation sequence diagram .....  | 87  |
| 4.6 Calendar sequence diagram .....    | 91  |
| 4.7 Profile sequence diagram .....     | 92  |
| 5. State machine diagram .....         | 95  |
| 6. User interface prototype .....      | 98  |
| 7. Implementation requirements .....   | 120 |
| 8. Glossary .....                      | 121 |
| 9. References .....                    | 122 |

= Authors for each section =

| <b>Sections.</b>            | <b>Authors</b> |
|-----------------------------|----------------|
| Introduction                | 김수현, 정은지, 황윤주  |
| Use case Analysis           | 김수현, 황윤주       |
| Class Diagram               | 정은지, 황윤주, 정명기  |
| Seqeunce Diagram            | 김수현            |
| State Machine Diagram       | 정은지            |
| User Interface Prototype    | 정은지            |
| Implementation Requirements | 김수현, 황윤주       |
| Glossary                    | 김수현, 황윤주       |
| References                  | 김수현            |

## 1. Introduction

본 문서는 Design Phase의 산출물로서, 우리 팀이 개발하고자 하는 시스템인 MeetMate 시스템의 구현 사항에 대한 명세를 포함하고 있다. 문서에서는 Requirement Analysis Phase에서 작성된 SRS(요구사항 명세서)를 기반으로 MeetMate의 기능적 요구사항을 구현하기 위한 시스템 설계를 다양한 관점에서 제시한다. 각 섹션은 시스템의 특정 측면에 대한 자세한 설계 내용을 담고 있다.

Use case analysis에서는 사용자의 관점에서 MeetMate가 제공하는 기능을 자세히 기술한다. 이 단계에서는 시스템의 사용자들이 원하는 기능과, 기능 간의 관계를 Use Case Diagram을 통해 나타내고, Use Case Description을 통해 각 Use Case가 수행하는 작업을 상세히 설명한다. 이를 통해 사용자의 요구를 파악하고, 시스템의 기능적 요구사항에 대한 이해를 제공한다.

Class Diagram은 MeetMate의 구조적인 관점에서의 설계를 제공한다. Use case analysis를 통해 식별된 사용자 관점의 기능들을 구현하는 데 필요한 객체들을 정의하고, 그를 구성하는 관계, 속성, 메소드 등을 제시한다.

Sequence Diagram에서는 작성된 Use case Diagram을 MeetMate의 동적 관점에서 묘사하여, 사용자와 시스템 간의 상호 작용을 시간의 흐름에 따라 나열하였다. 이때 사용되는 객체와 메시지는 모두 Class Diagram에서 제시된 관계, 속성, 메소드와 일치한다.

State machine diagram은 클라이언트 측 소프트웨어인 안드로이드 애플리케이션의 설계를 중점적으로 다루며 Server의 관점에서 Class diagram on Server Side을 작성하였고, Client 측 관점에서 Class diagram on Client Side을 작성하였다.

User interface prototype은 UI와 UI 안의 각 구성요소를 설명하였다. 사용자 입장과 관리자 입장에서 바라본 애플리케이션의 기능을 상세히 설명하였다.

## 2. Use case analysis

이 섹션에서는 Use Case Diagram과 각 Use Case에 대한 자세한 설명인 Use Case Descriptions을 제공한다. Requirement Analysis Phase에서 작성된 SRS(요구사항 명세서)를 기반으로 작성되었으며, StarUML을 활용하여 각 Use Case를 시각화하였다.

Use case diagram에 관한 고려 사항은 다음과 같다.

### 1. MeetMate 시스템의 세 가지 주요 사용자 유형

- General User (개인 사용자)
- Co. User (기업 사용자)
- Administrator (관리자)

모든 사용자는 회원가입을 제외한 MeetMate의 모든 기능을 활용하기 위해서는 '로그인 상태'여야 한다. 회원가입은 로그인 이전에 수행될 수 있다. DB 서버는 시스템 구현의 기술적 측면이므로 Actor로 고려하지 않았으며 문서에서 말하는 Administrator는 시스템 내부 관리자를 의미한다.

### 2. 기능 표현의 묶음

가독성을 고려하여 CRUD나 기본 연산 기능은 각각 표시하지 않고, "Manage"로 묶어 표현하였다. "Manage"로 시작하는 각 Use Case는 해당 분야(댓글, 프로필)와 관련된 연산이 포함된 Use Case이다. 세부 기능의 내용은 Use Case Description에서 자세히 설명한다.

### 3. View post list 확장 관계

"View post list"의 확장 관계인 "View entire post," "View category post," "View area post"은 글 목록 열람 기능은 같으나, 접근 방법이 각각 다르므로 별도로 표현하였다. 각 Use Case는 볼 수 있는 글의 범위가 다르지만, 실제로 글 목록을 보는 기능은 공통으로 포함되어 있다.

Use case description에 관한 고려 사항은 다음과 같다.

### 1. 기능 표현의 묶음

Manage로 묶어 표현된 use case의 CRUD 연산은 Extension scenarios에서 자세히 설명한다.

### 2. Primary Actor 구분

Primary Actor에 "User, Admin"이라고 적혀있다면, 해당 기능은 사용자와 관리자가 모두 사용한 것을 의미한다. 동일한 기능이라도 사용하는 경우가 다를 경우, User와 Admin을 분리하여 설명한다. 같은 경우라면 User와 Admin 모두 "User"로 동일시하여 설명한다.

### 3. User 유형에 따른 Use Case #1, #6

"Register"와 "Write post"에선 User를 일반 사용자와 기업 사용자로 나누어 설명하고, 사용자별 Extension Scenarios를 작성하였다.

### 4. Use Case #4 View post list

"View post list"의 확장 관계인 "View entire post," "View category post," "View area post"를 함께 서술하며, Extension Scenarios에서 차이점(접근 방식)을 설명한다.

### 5. Use Case #5 View post

"View post"는 글 보기 기능에 대한 Use Case이지만, 사용자는 글 보기 화면에서 댓글도 볼 수 있으므로 Extension Scenarios에서 함께 설명한다.

### 6. Use Case #6 Write post

Use case #6 Write post에서 작성한 글 내용은 Use case #12와 Use case #15로 이어진다. 자세한 내용은 description에서 설명한다.

### 7. Use Case #12~14

Use case #12~14는 모두 초대장 관련 기능으로, 접근 방법이 달라 Manage로 묶지 않고 따로 표현하였다.

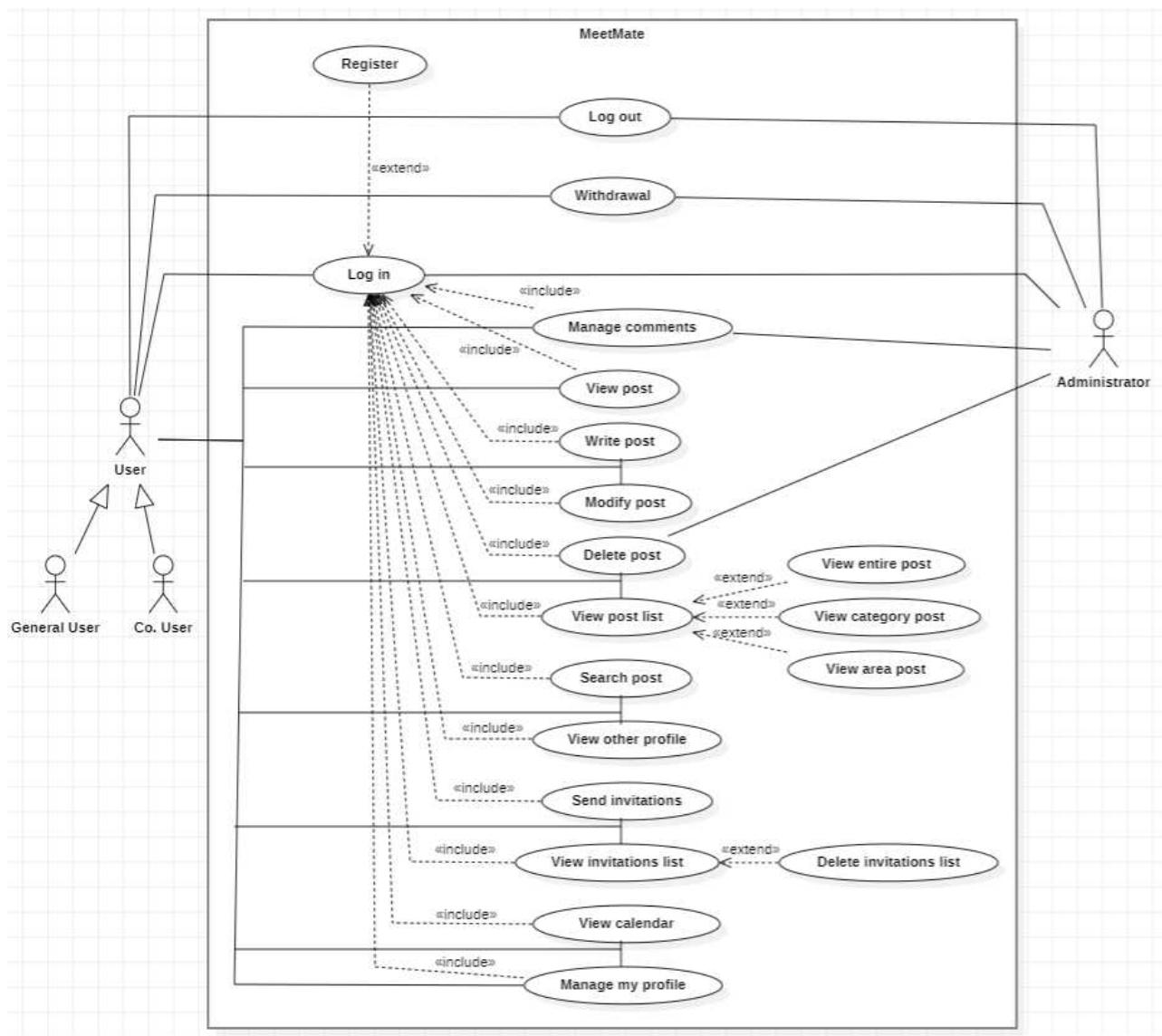
## 8. Use Case #15 View calendar

View calendar 화면에 나타나는 일정들은 모두 Use Case #13을 통해 시스템에서 자동으로 추가되는 일정이다.

## 9. Use Case Withdrawal

Use Case Withdrawal은 Use Case #17과 #18로 구분하여 설명한다. #17은 사용자가 직접 탈퇴할 경우, #18은 관리자가 사용자를 탈퇴시키는 경우이다.

### 2.1) Use case diagram



[그림 2-1] MeetMate의 Use case diagram

| #  | <b>Use case</b>        | <b>Summary</b>                                 |
|----|------------------------|--|
| 1  | Register               | 사용자가 시스템 사용을 위해 회원 등록을 한다.                     |
| 2  | Login                  | 사용자, 시스템 관리자가 MeetMate 시스템을 이용하기 위해 시스템에 접속한다. |
| 3  | Logout                 | 사용자, 시스템 관리자가 시스템에서 로그아웃한다.                    |
| 4  | View post list         | 사용자가 글 목록을 열람한다.                               |
| 5  | View Post              | 사용자가 특정 글을 열람한다.                               |
| 6  | Write Post             | 사용자가 글을 작성한다.                                  |
| 7  | Modify Post            | 사용자가 작성한 글을 수정한다.                              |
| 8  | Delete Post            | 사용자가 작성한 글을 삭제한다.                              |
| 9  | Manage Comments        | 사용자는 시스템을 이용해 댓글을 관리한다.                        |
| 10 | Search Post            | 사용자가 원하는 카테고리를 검색한다.                           |
| 11 | View Other Profile     | 사용자가 다른 사용자의 프로필을 본다.                          |
| 12 | Send Invitation        | 사용자가 다른 사용자에게 모임 초대장을 보낸다.                     |
| 13 | View Invitation List   | 사용자가 받은 초대장 목록을 열람한다.                          |
| 14 | Delete Invitation List | 사용자가 받은 초대장들을 모두 삭제한다.                         |
| 15 | View Calendar          | 사용자가 캘린더를 열람하여, 모임 정보를 확인한다.                   |
| 16 | Manage My Profile      | 사용자가 자신의 프로필을 관리한다.                            |
| 17 | Withdrawal             | 사용자가 시스템을 탈퇴한다.                                |
| 18 | Withdrawal             | 관리자가 사용자를 탈퇴시킨다.                               |

| Use case #1 : Register  |   |
|-------------------------|---|
| GENERAL CHARACTERISTICS |   |
| Summary                 | 시스템의 기능을 처음 이용하려 할 때 필요한 절차로, user는가 시스템 사용에 앞서 회원가입을 한다. |
| Scope                   | MeetMate  |
| Level                   | User level  |
| Author                  | 김수현   |
| Last Update             | 2023.11.01  |
| Status                  | Analysis  |
| Primary Actor           | User  |
| Preconditions           | 사용자는 시스템을 설치하여 실행한 상태여야 한다.                               |
| Trigger                 | 시스템 실행 후 로그인 화면에서 회원가입 버튼을 누를 때                           |
| Success Post Condition  | 사용자는 회원가입에 성공한다.  |
| Failed Post Condition   | 사용자는 회원가입에 실패한다.  |

| MAIN SUCCESS SCENARIO |   |
|-----------------------|---|
| Step                  | Action  |
| S                     | 사용자가 시스템에 회원가입 버튼을 누를 때 시작된다.                         |
| 1                     | 사용자는 로그인 화면에서 회원가입 버튼을 누르고 회원 종류를 선택한다.               |
| 2                     | 시스템은 회원가입 하려는 사용자의 회원 종류에 맞게 회원가입 정보 입력 화면을 출력한다.     |
| 3                     | 사용자는 ID, PW, 이름, 닉네임, 취미 카테고리, 위치를 입력하고 회원가입 버튼을 누른다. |
| 4                     | 시스템은 회원가입 성공 여부를 판단한다.                                |
| 5                     | 이 use case는 DB에 사용자의 정보가 저장된 다음, 로그인 화면으로 넘어갈 때 끝난다.  |

| EXTENSION SCENARIOS |  |
|---------------------|--|
| Step                | Branching Action   |
| 1                   | 1a. 개인 회원으로 회원가입<br>…1a1. 사용자는 체크 박스를 누르지 않고 개인 회원으로 회원가입 한다.<br>…1a2. 시스템은 회원가입 정보 입력 화면을 출력한다.<br>…1a3. 사용자는 ID, PW, 이름, 닉네임, 취미 카테고리, 위치(지역구)를 입력하고 회원가입을 한다. (Use Case #1-3) |
| 1                   | 1b. 기업 회원으로 회원가입<br>…1b1. 사용자는 체크 박스를 눌러 기업 회원으로 회원가입 한다.<br>…1b2. 시스템은 회원가입 정보 입력 화면을 출력한다.<br>…1b3. 사용자는 ID, PW, 이름, 기업명, 기업 카테고리, 기업 위치(지역구)를 입력하고 회원가입을 한다. (Use Case #1-3)  |

|   |   |
|---|---|
| 3 | <p>3a. 중복된 ID 입력</p> <ul style="list-style-type: none"> <li>…3a1. ID 중복 에러메시지를 출력한다.</li> <li>…3a2. 회원 정보 입력 화면으로 돌아간다.</li> </ul> <p>3b. 닉네임 규정을 준수하지 않은 잘못된 닉네임 입력 (중복 금지, 8자 이내, 특수문자 제외)</p> <ul style="list-style-type: none"> <li>…3b1. 닉네임 에러메시지를 출력한다.</li> <li>…3b2. 회원 정보 입력 화면으로 돌아간다.</li> </ul> <p>3c. 개인 회원이 선택한 취미 카테고리가 3개일 경우</p> <ul style="list-style-type: none"> <li>…3c1. 취미 선택 기능이 비활성화 되고, 회원 정보 입력 화면을 유지한다.</li> </ul> <p>3d. 기업 회원이 선택한 취미 카테고리가 1개일 경우</p> <ul style="list-style-type: none"> <li>…3d1. 취미 선택 기능이 비활성화 되고, 회원 정보 입력 화면을 유지한다.</li> </ul> <p>3e. 선택한 취미 카테고리가 없을 경우</p> <ul style="list-style-type: none"> <li>…3e1. null 에러메시지를 출력한다.</li> <li>…3e2. 회원 정보 입력 화면으로 돌아간다.</li> </ul> <p>3f. 개인 회원이 선택한 위치(지역구)가 3개일 경우</p> <ul style="list-style-type: none"> <li>…3f1. 취미 선택 기능이 비활성화 되고, 회원 정보 입력 화면을 유지한다.</li> </ul> <p>3g. 기업 회원이 선택한 위치(지역구)가 1개일 경우</p> <ul style="list-style-type: none"> <li>…3g1. 취미 선택 기능이 비활성화 되고, 회원 정보 입력 화면을 유지한다.</li> </ul> <p>3h. 선택한 위치가 없을 경우</p> <ul style="list-style-type: none"> <li>…3h1. null 에러메시지를 출력한다.</li> <li>…3h2. 회원 정보 입력 화면으로 돌아간다.</li> </ul> |
|---|---|

#### RELATED INFORMATION

|               |             |
|---------------|-------------|
| Performance   | ≤ 3 seconds |
| Frequency     | 사용자당 1번     |
| <Concurrency> | 제한 없음       |
| Due Date      | 2023.11.30  |

| Use case #2 : Log in    |  |
|-------------------------|--|
| GENERAL CHARACTERISTICS |  |
| Summary                 | 사용자, 시스템 관리자가 MeetMate 시스템을 이용하기 위해 시스템에 접속한다. |
| Scope                   | MeetMate                                       |
| Level                   | User level                                     |
| Author                  | 김수현  |
| Last Update             | 2023.11.01                                     |
| Status                  | Analysis                                       |
| Primary Actor           | User   |
| Preconditions           | 회원 등록이 되어있고, 로그인이 되지 않은 상태여야 한다.               |
| Trigger                 | 로그인 화면에서 ID, 비밀번호를 입력 후 로그인 버튼을 누를 때           |
| Success Post Condition  | 사용자는 로그인에 성공하여 시스템의 모든 기능을 사용할 수 있다.           |
| Failed Post Condition   | 사용자는 로그인에 실패하여 회원가입을 제외한 시스템의 모든 기능을 사용할 수 없다. |

| MAIN SUCCESS SCENARIO |  |
|-----------------------|--|
| Step                  | Action                                     |
| S                     | 사용자가 시스템에 로그인할 때 시작된다.                     |
| 1                     | 사용자는 로그인 화면에서 ID와 비밀번호를 입력한 후 로그인 버튼을 누른다. |
| 2                     | 시스템은 사용자의 정보를 검사하여 로그인 성공 여부를 판단한다.        |
| 3                     | 시스템은 등록된 사용자라면 로그인을 성공시킨 뒤 홈 화면을 출력한다.     |
| 4                     | 이 use case는 로그인이 성공하면 끝난다.                 |

| EXTENSION SCENARIOS |  |
|---------------------|--|
| Step                | Branching Action   |
| 3                   | 3a. 등록되지 않은 정보를 입력해 로그인 실패<br>…3a1. 로그인 에러 메시지를 출력한다.<br>…3a2. ID 입력 단계로 돌아간다. (Use case #2-1) |

| RELATED INFORMATION |               |
|---------------------|---------------|
| Performance         | ≤ 3 seconds   |
| Frequency           | 사용자당 하루 평균 1번 |
| <Concurrency>       | 제한 없음         |
| Due Date            | 2023.11.30    |

| Use case #3 : Log out   |  |
|-------------------------|--|
| GENERAL CHARACTERISTICS |  |
| Summary                 | 사용자, 시스템 관리자가 시스템에서 로그아웃한다.                              |
| Scope                   | MeetMate   |
| Level                   | User level   |
| Author                  | 김수현  |
| Last Update             | 2023.11.01   |
| Status                  | Analysis   |
| Primary Actor           | User   |
| Preconditions           | 사용자는 로그인이 된 상태여야 한다.<br>Mypage 탭을 눌러 프로필 화면이 열린 상태여야 한다. |
| Trigger                 | 로그인된 상태에서 Mypage 탭에서 로그아웃 버튼을 누를 때                       |
| Success Post Condition  | 사용자는 로그아웃에 성공한다.   |
| Failed Post Condition   | 사용자는 로그아웃에 실패해 로그인 상태를 유지한다.                             |

| MAIN SUCCESS SCENARIO |  |
|-----------------------|--|
| Step                  | Action                                       |
| S                     | 사용자가 로그인한 상태에서 Mypage 화면 로그아웃 버튼을 누를 때 시작된다. |
| 1                     | 사용자는 Mypage 탭에서 로그아웃 버튼을 누른다.                |
| 2                     | 시스템은 로그아웃 여부를 묻는 로그아웃 확인창을 출력한다.             |
| 3                     | 사용자는 로그아웃 확인창에서 “예” 버튼을 누른다.                 |
| 4                     | 시스템은 로그아웃 성공 여부를 판단한다.                       |
| 5                     | 시스템은 로그아웃에 성공하면 로그인 화면을 출력한다.                |
| 6                     | 이 use case는 로그아웃이 성공하면 끝난다.                  |

| EXTENSION SCENARIOS |  |
|---------------------|--|
| Step                | Branching Action   |
| 3                   | 3a. 통신에 실패해 로그아웃 실패<br>...3a1. 로그아웃 확인창이 사라지고 Mypage 화면으로 돌아간다. (Use case #3-1)<br>3b. 로그아웃 확인창에서 “아니오” 버튼을 눌러 로그아웃 실패<br>...3b1. 로그아웃 확인창이 사라지고 Mypage 화면으로 돌아간다. (Use case #3-1) |

| RELATED INFORMATION |               |
|---------------------|---------------|
| Performance         | ≤ 3 seconds   |
| Frequency           | 사용자당 하루 평균 1번 |
| <Concurrency>       | 제한 없음         |
| Due Date            | 2023.11.30    |

**Use case #4 : View post list**
**GENERAL CHARACTERISTICS**

|                        |   |
|------------------------|---|
| Summary                | 사용자가 글 목록을 본다. 흄·카테고리·주변취미 탭을 이용해 원하는 범위의 글 목록을 볼 수 있다. |
| Scope                  | MeetMate  |
| Level                  | User level  |
| Author                 | 김수현   |
| Last Update            | 2023.11.01  |
| Status                 | Analysis  |
| Primary Actor          | User  |
| Preconditions          | 사용자는 시스템에 로그인한 상태여야 한다. 통신할 수 있어야 한다.                   |
| Trigger                | 로그인된 상태에서 흄·카테고리·주변취미 탭 중 하나를 눌렀을 때                     |
| Success Post Condition | 각 탭별로 다른 범위의 글 목록이 나오며, 사용자는 본인이 원하는 범위의 글 목록을 볼 수 있다.  |
| Failed Post Condition  | 글 목록 화면이 나오지 않는다.                                       |

**MAIN SUCCESS SCENARIO**

| Step | Action   |
|------|--|
| S    | 사용자가 시스템에 로그인해 흄, 카테고리, 주변취미 탭 중 하나를 누를 때 시작된다.                |
| 1    | 사용자는 원하는 탭을 선택하여 누른다.  |
| 2    | 시스템은 탭 설정에 맞는 글 목록 화면을 출력한다.                                   |
| 3    | 사용자는 글 목록에서 글 제목, 글 내용(앞에서 10글자), 작성 시간, 글 작성자의 닉네임을 확인할 수 있다. |

**EXTENSION SCENARIOS**

| Step | Branching Action  |
|------|---|
| 1    | 1a. 흄 탭 누르기<br>…1a1. 사용자는 취미 카테고리, 지역 상관없이 모든 글 목록을 볼 수 있다.<br>1b. 카테고리 탭 누르기<br>…1b1. 사용자는 지역 상관없이 사용자가 선택한 취미 카테고리에 속한 글 목록을 볼 수 있다.<br>1c. 주변취미 탭 누르기<br>…1c1. 사용자는 취미 카테고리 상관없이 사용자가 선택한 위치에 속한 글 목록을 볼 수 있다. |

**RELATED INFORMATION**

|               |             |
|---------------|-------------|
| Performance   | ≤ 3 seconds |
| Frequency     | 알 수 없음      |
| <Concurrency> | 제한 없음       |
| Due Date      | 2023.11.30  |

**Use case #5 : View post**
**GENERAL CHARACTERISTICS**

|                        |  |
|------------------------|--|
| Summary                | 사용자가 글을 본다. 글 목록 화면에서 원하는 글을 눌러 볼 수 있다.  |
| Scope                  | MeetMate   |
| Level                  | User level   |
| Author                 | 김수현  |
| Last Update            | 2023.11.18   |
| Status                 | Analysis   |
| Primary Actor          | User   |
| Preconditions          | 사용자는 시스템에 로그인한 상태여야 한다.<br>글 목록 보기 화면(Use case #4 View post list)이 열린 상태여야 한다. |
| Trigger                | 글 목록 화면에서 글을 눌렀을 때   |
| Success Post Condition | 사용자는 본인이 원하는 글을 볼 수 있다.  |
| Failed Post Condition  | 글 보기 화면이 나오지 않는다.  |

**MAIN SUCCESS SCENARIO**

| Step | Action  |
|------|---|
| S    | 사용자가 글 목록 보기 화면에서 글을 누른다.                                     |
| 1    | 사용자는 원하는 글을 선택하여 누른다.   |
| 2    | 시스템은 사용자가 선택한 글 화면을 출력한다.                                     |
| 3    | 사용자는 글 화면에서 글 제목, 카테고리, 위치, 작성 시간, 글 작성자의 닉네임, 댓글을 확인 할 수 있다. |

**EXTENSION SCENARIOS**

| Step | Branching Action  |
|------|---|
| 3    | 3a. 글 보기 (User)<br>…3a1. 사용자는 글 보기 화면에서 자신이 작성한 글 내용을 확인한다.<br>3b. 글 보기 (Other user)<br>…3b1. 사용자는 글 보기 화면에서 다른 사용자가 작성한 글 내용을 확인한다.<br>…3b2. 글 제목, 내용, 글 작성자의 닉네임, 작성 시간, 댓글을 볼 수 있다.<br>…3b3. 사용자는 글 보기 화면에서 다른 작성자의 닉네임을 눌러 상호 작용 창을 열 수 있다.<br>…3b4. 사용자는 상호 작용 창에서 Use case #11과 Use case #12를 진행할 수 있다.<br>3c. 댓글 보기<br>…3c1. 사용자는 댓글 화면에서 자신이 작성한 댓글 내용을 확인할 수 있다.<br>…3c2. 사용자는 댓글 화면에서 다른 사용자가 작성한 댓글 내용을 확인할 수 있다.<br>…3c3. 사용자는 댓글 화면에서 다른 작성자의 닉네임을 눌러 상호 작용 창을 열 수 있다.<br>…3c4. 사용자는 상호작용창에서 Use case #11과 Use case #12를 진행할 수 있다. |

| RELATED INFORMATION |             |
|---------------------|-------------|
| Performance         | ≤ 3 seconds |
| Frequency           | 알 수 없음      |
| <Concurrency>       | 제한 없음       |
| Due Date            | 2023.11.30  |

**Use case #6 : Write post**
**GENERAL CHARACTERISTICS**

|                               |   |
|-------------------------------|---|
| <b>Summary</b>                | 사용자가 글을 작성한다.                                       |
| <b>Scope</b>                  | MeetMate  |
| <b>Level</b>                  | User level  |
| <b>Author</b>                 | 김수현   |
| <b>Last Update</b>            | 2023.11.18  |
| <b>Status</b>                 | Analysis  |
| <b>Primary Actor</b>          | User  |
| <b>Preconditions</b>          | 글 목록 보기 화면(Use case #4 View post list)이 열린 상태여야 한다. |
| <b>Trigger</b>                | 사용자가 글을 작성하려고 할 때                                   |
| <b>Success Post Condition</b> | 사용자는 글을 작성할 수 있다.                                   |
| <b>Failed Post Condition</b>  | 글쓰기창이 나오지 않는다.                                      |

**MAIN SUCCESS SCENARIO**

| Step | Action                                    |
|------|---|
| S    | 사용자는 Use case #4 화면에서 글 쓰기 기능을 사용하려 한다.   |
| 1    | 사용자는 글 목록 보기 화면에서 글쓰기 버튼을 누른다.            |
| 2    | 시스템은 글을 쓸 수 있는 글쓰기 창을 제공한다.               |
| 3    | 사용자는 취미 카테고리, 지역구를 선택하고 글 제목, 글 내용을 작성한다. |
| 4    | 사용자는 글을 쓰고 등록 버튼을 누른다.                    |
| 5    | 시스템은 글을 저장하고 화면에 출력한다.                    |

**EXTENSION SCENARIOS**

| Step | Branching Action  |
|------|---|
| 3    | 3a. 취미 카테고리, 위치(지역구)를 선택하지 않을 경우<br>…3a1. 시스템은 글쓰기 창에서 기본 취미 카테고리, 위치(지역구)를 출력한다.<br>…3a2. 사용자가 따로 선택하지 않으면 기본 취미 카테고리와 위치로 글이 작성된다.<br>3b. 글 제목을 입력하지 않으면 글 등록에 실패한다.<br>…3b1. 제목 에러 메시지를 출력한다.<br>…3b2. 글쓰기 단계(Use case #6-3)로 돌아간다. 이때, 작성해 둔 것들은 사라지지 않는다.<br>3c. 글 내용을 모두 입력하지 않을 경우 글 등록에 실패한다.<br>…3c1. 내용 에러 메시지를 출력한다.<br>…3c2. 글쓰기 단계(Use case #6-3)로 돌아간다. 이때, 작성해 둔 것들은 사라지지 않는다. |
| 5    | 5a. 개인 회원이 글을 3개 이상 등록할 경우<br>…5a1. 개인 회원은 글을 하루에 3개만 작성할 수 있도록 제한한다.   |

|  |  |
|--|--|
|  | <p>…5a2. 개인 회원이 글을 더 쓰기 위해 글쓰기 버튼을 눌렀을 경우, 시스템은 글 개수 초과 에러 메시지를 출력한다.</p> <p>5b. 기업 회원이 글을 1개 이상 등록할 경우</p> <p>…5b1. 기업 회원은 글을 하루에 1개만 작성할 수 있도록 제한한다.</p> <p>…5b2. 기업 회원이 글을 더 쓰기 위해 글쓰기 버튼을 눌렀을 경우, 시스템은 글 개수 초과 에러 메시지를 출력한다.</p> |
|--|--|

| RELATED INFORMATION |             |
|---------------------|-------------|
| Performance         | ≤ 3 seconds |
| Frequency           | 알 수 없음      |
| <Concurrency>       | 제한 없음       |
| Due Date            | 2023.11.30  |

**Use case #7 : Modify post**
**GENERAL CHARACTERISTICS**

|                               |                                     |
|-------------------------------|-------------------------------------|
| <b>Summary</b>                | 사용자가 작성한 글을 수정한다.                   |
| <b>Scope</b>                  | MeetMate                            |
| <b>Level</b>                  | User level                          |
| <b>Author</b>                 | 김수현                                 |
| <b>Last Update</b>            | 2023.11.18                          |
| <b>Status</b>                 | Analysis                            |
| <b>Primary Actor</b>          | User                                |
| <b>Preconditions</b>          | 글 보기 화면(Use case #5)이 열린 상태여야 한다.   |
| <b>Trigger</b>                | 사용자가 글을 수정하려고 할 때                   |
| <b>Success Post Condition</b> | 사용자가 수정을 시도한 글이 시스템에서 수정되어 업데이트 된다. |
| <b>Failed Post Condition</b>  | 사용자는 글을 수정할 수 없다.                   |

**MAIN SUCCESS SCENARIO**

| Step | Action                                  |
|------|---|
| S    | 사용자는 Use case #5 화면에서 글 수정 기능을 사용하려 한다. |
| 1    | 사용자는 글 보기 화면에서 수정 버튼을 누른다.              |
| 2    | 시스템은 사용자가 작성했던 글 내용이 그대로 있는 글쓰기창을 제공한다. |
| 3    | 사용자는 글을 수정하고 수정 버튼을 누른다.                |
| 4    | 시스템은 수정된 글을 저장하고 수정된 글을 출력한다.           |

**EXTENSION SCENARIOS**

| Step | Branching Action  |
|------|---|
| 1    | 1a. 다른 사용자의 글 화면에서 수정 버튼을 누를 경우<br>…1a1. 수정 에러 메시지를 출력한다.<br>…1a2. 글 보기 화면(Use case #5)도 돌아간다.  |
| 3    | 3a. 사용자가 수정 과정에서 글 제목을 입력하지 않으면 글 등록에 실패한다.<br>…3a1. 제목 에러 메시지를 출력한다.<br>…3a2. 글쓰기 단계(Use case #6-3)로 돌아간다. 이때, 작성해 둔 것들은 사라지지 않는다.<br>3b. 사용자가 수정 과정에서 글 내용을 모두 입력하지 않으면 글 등록에 실패한다.<br>…3b1. 내용 에러 메시지를 출력한다.<br>…3b2. 글쓰기 단계(Use case #6-3)로 돌아간다. 이때, 작성해 둔 것들은 사라지지 않는다. |

**RELATED INFORMATION**

|                    |             |
|--------------------|-------------|
| <b>Performance</b> | ≤ 3 seconds |
|--------------------|-------------|

|               |               |
|---------------|---------------|
| Frequency     | 사용자당 하루 평균 1번 |
| <Concurrency> | 제한 없음         |
| Due Date      | 2023.11.30    |

**Use case #8 : Delete post**
**GENERAL CHARACTERISTICS**

|                        |                                   |
|------------------------|-----------------------------------|
| Summary                | 사용자가 작성한 글을 삭제한다.                 |
| Scope                  | MeetMate                          |
| Level                  | User level                        |
| Author                 | 김수현                               |
| Last Update            | 2023.11.18                        |
| Status                 | Analysis                          |
| Primary Actor          | User, Admin                       |
| Preconditions          | 글 보기 화면(Use case #5)이 열린 상태여야 한다. |
| Trigger                | 사용자가 글을 삭제하려고 할 때                 |
| Success Post Condition | 사용자가 삭제를 시도한 글이 시스템에서 삭제된다.       |
| Failed Post Condition  | 사용자는 글을 삭제할 수 없다.                 |

**MAIN SUCCESS SCENARIO**

| Step | Action                                      |
|------|---|
| S    | 사용자는 Use case #5 화면에서 글 삭제 기능을 사용하려 한다.     |
| 1    | 사용자는 글 보기 화면에서 삭제 버튼을 누른다.                  |
| 2    | 시스템은 글을 삭제하고 글 목록 보기 화면(Use case #4)을 출력한다. |

**EXTENSION SCENARIOS**

| Step | Branching Action  |
|------|---|
| 1    | 1a. 다른 사용자의 글 화면에서 삭제 버튼을 누를 경우 (User)<br>…1a1. 삭제 에러 메시지를 출력한다.<br>…1a2. 글 보기 화면(Use case #5)도 돌아간다.<br>1b. 다른 사용자의 글 화면에서 삭제 버튼을 누를 경우 (Admin)<br>…1b1. Admin은 모집 글의 규정에 맞지 않는 글이 있으면 삭제 버튼을 누른다.<br>…1b2. Use case #8-2를 수행한다. |

**RELATED INFORMATION**

|               |               |
|---------------|---------------|
| Performance   | ≤ 3 seconds   |
| Frequency     | 사용자당 하루 평균 1번 |
| <Concurrency> | 제한 없음         |
| Due Date      | 2023.11.30    |

**Use case #9 : Manage comments**
**GENERAL CHARACTERISTICS**

|                               |   |
|-------------------------------|---|
| <b>Summary</b>                | 사용자는 시스템을 이용해 댓글을 관리한다.                                     |
| <b>Scope</b>                  | MeetMate  |
| <b>Level</b>                  | User level  |
| <b>Author</b>                 | 김수현   |
| <b>Last Update</b>            | 2023.11.18  |
| <b>Status</b>                 | Analysis  |
| <b>Primary Actor</b>          | User, Admin   |
| <b>Preconditions</b>          | 사용자는 로그인한 상태여야 한다.<br>사용자는 글 보기 화면 속에 있는 댓글 화면이 열린 상태여야 한다. |
| <b>Trigger</b>                | 사용자가 시스템을 이용해 댓글 관련 작업을 하려고 할 때                             |
| <b>Success Post Condition</b> | 사용자는 시스템을 이용해서 댓글 관련 기능을 다양하게 사용할 수 있다.                     |
| <b>Failed Post Condition</b>  | 사용자는 시스템을 이용해서 댓글 관련 기능을 사용할 수 없다.                          |

**MAIN SUCCESS SCENARIO**

| Step | Action   |
|------|--|
| S    | 사용자는 Use case #5 View post의 글 보기 화면에서 댓글 관련 기능을 사용하려 한다. |
| 1    | 사용자는 댓글 화면을 통해 사용자들의 닉네임, 작성 시간, 작성한 댓글 내용을 볼 수 있다.      |
| 2    | 사용자는 댓글창을 누른다.   |
| 3    | 사용자는 댓글을 작성하고 등록 버튼을 누른다.                                |
| 4    | 시스템은 댓글을 저장하고 화면에 출력한다.                                  |
| 5    | 단계 4 이후로 사용자는 원할 때 댓글을 수정하거나 삭제할 수 있다.                   |

**EXTENSION SCENARIOS**

| Step | Branching Action  |
|------|---|
| 1    | 1a. 댓글 속 다른 사용자의 닉네임 누르기<br>…1a1. 댓글 속 다른 사용자의 닉네임을 눌러 상호 작용 창을 열 수 있다.<br>…1a2. 사용자는 상호작용창에서 Use case #11과 Use case #12를 진행할 수 있다. |
| 3    | 3a. 댓글 작성<br>…3a1. 사용자는 모임에 참가 의사를 표현한 댓글을 작성한다.<br>…3a2. 댓글 등록에 실패할 경우, 댓글 입력창을 다시 띄워 댓글 작성 단계로 돌아간다.                              |
| 5    | 5a. 댓글 수정<br>…5a1. 사용자(댓글 작성자)는 댓글 화면에서 수정 버튼을 누른다.<br>…5a2. 시스템은 사용자가 작성했던 댓글 내용을 띄운 댓글창을 출력한다.                                    |

|  |   |
|--|---|
|  | <p>…5a3. 사용자는 댓글 내용을 수정하고 수정 버튼을 누른다.</p> <p>…5a4. 시스템은 수정된 댓글을 저장하고 수정된 댓글을 출력한다.</p> <p>5b. 댓글 삭제 (User)</p> <p>…5b1. 사용자(댓글 작성자)는 댓글 화면에서 삭제 버튼을 누른다.</p> <p>…5b2. 시스템은 댓글을 삭제하고 댓글 화면을 출력한다.</p> <p>5c. 댓글 삭제 (Admin)</p> <p>…5c1. Admin(관리자)은 글 내용과 맞지 않은 댓글이 있을 경우 확인 후 삭제 버튼을 누른다.</p> <p>…5c2. 시스템은 댓글을 삭제하고 댓글 화면을 출력한다.</p> |
|--|---|

#### RELATED INFORMATION

|               |               |
|---------------|---------------|
| Performance   | ≤ 3 seconds   |
| Frequency     | 사용자당 하루 평균 2번 |
| <Concurrency> | 제한 없음         |
| Due Date      | 2023.11.30    |

**Use case #10 : Search post**
**GENERAL CHARACTERISTICS**

|                               |                                   |
|-------------------------------|-----------------------------------|
| <b>Summary</b>                | 사용자가 원하는 카테고리를 검색한다.              |
| <b>Scope</b>                  | MeetMate                          |
| <b>Level</b>                  | User level                        |
| <b>Author</b>                 | 김수현                               |
| <b>Last Update</b>            | 2023.11.18                        |
| <b>Status</b>                 | Analysis                          |
| <b>Primary Actor</b>          | User                              |
| <b>Preconditions</b>          | 시스템에 로그인하여 화면을 켜둔 상태여야 한다.        |
| <b>Trigger</b>                | 상단 바에 있는 검색 버튼을 누를 때              |
| <b>Success Post Condition</b> | 사용자는 선택한 카테고리에 해당하는 글 목록을 볼 수 있다. |
| <b>Failed Post Condition</b>  | 사용자는 카테고리에 해당하는 글 목록을 볼 수 없다.     |

**MAIN SUCCESS SCENARIO**

| Step | Action                        |
|------|-------------------------------|
| S    | 사용자가 검색을 시도한다.                |
| 1    | 사용자는 시스템 상단 바에 있는 검색 버튼을 누른다. |
| 2    | 시스템은 카테고리 선택 화면을 출력한다.        |
| 3    | 사용자는 검색하고자 하는 카테고리를 선택한다.     |
| 4    | 시스템은 카테고리에 해당 되는 글 목록을 출력한다.  |

**EXTENSION SCENARIOS**

| Step | Branching Action  |
|------|---|
| 4    | 4a. 사용자가 뒤로가기 버튼을 누를 경우<br>…4a1. 검색 버튼을 누르기 전 화면으로 돌아간다.<br>4b. 사용자가 선택한 카테고리의 글이 없을 경우<br>…4b1. 시스템은 빈 화면을 출력한다. |

**RELATED INFORMATION**

|                            |               |
|----------------------------|---------------|
| <b>Performance</b>         | ≤ 3 seconds   |
| <b>Frequency</b>           | 사용자당 하루 평균 3번 |
| <b>&lt;Concurrency&gt;</b> | 제한 없음         |
| <b>Due Date</b>            | 2023.11.30    |

**Use case #11 : View other profile**
**GENERAL CHARACTERISTICS**

|                               |   |
|-------------------------------|---|
| <b>Summary</b>                | 다른 사용자의 프로필을 본다.  |
| <b>Scope</b>                  | MeetMate  |
| <b>Level</b>                  | User level  |
| <b>Author</b>                 | 김수현   |
| <b>Last Update</b>            | 2023.11.01  |
| <b>Status</b>                 | Analysis  |
| <b>Primary Actor</b>          | User  |
| <b>Preconditions</b>          | <p>다른 사용자의 닉네임을 누를 수 있는 화면이 열린 상태여야 한다.<br/>         (Use case #4~5)</p> <p>다른 사용자의 글이 최소 1개 이상 있어야 한다.</p> |
| <b>Trigger</b>                | 사용자가 다른 사용자의 닉네임을 눌러 프로필 보기 버튼을 누를 때  |
| <b>Success Post Condition</b> | 다른 사용자의 프로필 화면이 출력된다.   |
| <b>Failed Post Condition</b>  | 다른 사용자의 프로필 화면이 출력되지 않는다.   |

**MAIN SUCCESS SCENARIO**

| Step | Action   |
|------|--|
| S    | 사용자가 다른 사용자의 프로필을 본다.                                    |
| 1    | 사용자는 Use case #4~5 화면에서 다른 사용자의 닉네임을 누른다.                |
| 2    | 시스템은 다른 사용자와의 상호 작용 창을 화면에 출력한다.                         |
| 3    | 사용자는 상호 작용 창에서 다른 사용자의 닉네임, 취미, 위치, 가입일이 적힌 프로필을 볼 수 있다. |

**EXTENSION SCENARIOS**

| Step | Branching Action  |
|------|---|
| 3    | 3a. 사용자가 상호 작용 창을 닫고 싶을 경우<br>...3a1. 상호 작용 창의 바깥 화면을 눌러 Use case #4~5 화면으로 돌아간다. |

**RELATED INFORMATION**

|                            |                  |
|----------------------------|------------------|
| <b>Performance</b>         | $\leq$ 2 seconds |
| <b>Frequency</b>           | 사용자당 하루 평균 3번    |
| <b>&lt;Concurrency&gt;</b> | 제한 없음            |
| <b>Due Date</b>            | 2023.11.30       |

**Use case #12 : Send invitation**
**GENERAL CHARACTERISTICS**

|                               |   |
|-------------------------------|---|
| <b>Summary</b>                | 다른 사용자에게 모임 초대장을 보낸다.                         |
| <b>Scope</b>                  | MeetMate                                      |
| <b>Level</b>                  | User level                                    |
| <b>Author</b>                 | 김수현   |
| <b>Last Update</b>            | 2023.11.01                                    |
| <b>Status</b>                 | Analysis                                      |
| <b>Primary Actor</b>          | User  |
| <b>Preconditions</b>          | 사용자는 자신이 작성한 글 화면이 열린 상태여야 한다.                |
| <b>Trigger</b>                | 사용자가 자신의 글에서 다른 사용자의 닉네임을 눌러 초대장 보내기 버튼을 누를 때 |
| <b>Success Post Condition</b> | 다른 사용자에게 모임 초대장을 보낸다.                         |
| <b>Failed Post Condition</b>  | 다른 사용자에게 모임 초대장을 보내는 데 실패한다.                  |

**MAIN SUCCESS SCENARIO**

| Step | Action   |
|------|--|
| S    | 사용자가 다른 사용자에게 모임 초대장을 보낸다.                                       |
| 1    | 사용자는 Use case#5 화면에서 다른 사용자의 닉네임을 누른다.                           |
| 2    | 시스템은 다른 사용자와의 상호 작용 창을 화면에 출력한다.                                 |
| 3    | 사용자는 상호 작용 창에서 초대장 보내기 버튼을 누른다.                                  |
| 4    | 시스템은 사용자가 작성한 글 내용(Use case #6)을 DB에 저장한 후, 다른 사용자에게 초대장으로 전송한다. |
| 5    | 시스템은 초대장 전송 성공 메시지를 출력한다.  |
| 6    | Ol use case는 초대장 보내기에 성공하면 끝난다.                                  |

**EXTENSION SCENARIOS**

| Step | Branching Action  |
|------|---|
| 3    | 3a. 여러 사용자에게 초대장을 보낼 경우<br>…3a1. 다수의 사용자에게 초대장을 보내려면 사용자들의 닉네임을 각각 눌러 초대장을 전송해야 한다. |
| 5    | 5a. 초대장 전송에 실패한 경우<br>…5a1. 시스템은 초대장 보내기에 실패했다는 에러 메시지를 출력한다.                       |

**RELATED INFORMATION**

|                    |                  |
|--------------------|------------------|
| <b>Performance</b> | $\leq 2$ seconds |
|--------------------|------------------|

|               |            |
|---------------|------------|
| Frequency     | 알 수 없음     |
| <Concurrency> | 제한 없음      |
| Due Date      | 2023.11.30 |

**Use case #13 : View invitation list**
**GENERAL CHARACTERISTICS**

|                               |                                     |
|-------------------------------|-------------------------------------|
| <b>Summary</b>                | 사용자가 받은 초대장 목록을 본다.                 |
| <b>Scope</b>                  | MeetMate                            |
| <b>Level</b>                  | User level                          |
| <b>Author</b>                 | 김수현                                 |
| <b>Last Update</b>            | 2023.11.18                          |
| <b>Status</b>                 | Analysis                            |
| <b>Primary Actor</b>          | User                                |
| <b>Preconditions</b>          | 사용자는 로그인 되어 있어야 한다.                 |
| <b>Trigger</b>                | 다른 사용자에게 초대장을 받았을 때                 |
| <b>Success Post Condition</b> | 사용자는 받은 초대장 목록과 각각의 초대장 내용을 볼 수 있다. |
| <b>Failed Post Condition</b>  | 사용자는 받은 초대장 목록을 볼 수 없다.             |

**MAIN SUCCESS SCENARIO**

| Step | Action                              |
|------|-------------------------------------|
| S    | 사용자가 받은 초대장 목록을 본다.                 |
| 1    | 사용자는 시스템 상단에 있는 초대장 아이콘을 누른다.       |
| 2    | 시스템은 받은 초대장 목록 화면을 화면에 출력한다.        |
| 3    | 사용자는 초대장 목록 중 원하는 초대장을 누른다.         |
| 4    | 시스템은 사용자가 누른 초대장의 상세 정보를 출력한다.      |
| 5    | 사용자는 초대장의 상세 정보를 보고 모임 참석 여부를 결정한다. |

**EXTENSION SCENARIOS**

| Step | Branching Action   |
|------|--|
| 2    | 2a. 다른 사용자에게 받은 초대장이 없을 경우<br>…2a1. 시스템은 빈 초대장 목록 화면을 출력한다.  |
| 4    | 4a. 초대장 상세 정보 출력<br>…4a1. 초대장을 보낸 사용자(송신자) 닉네임과 초대장 내용 (모임 이름, 일정, 위치, 모임 카테고리, 오픈채팅 링크)을 본다.  |
| 5    | 5a. 초대장에 수락 버튼을 눌렀을 경우<br>…5a1. 시스템은 초대장 내용을 기반으로 수락된 모임 정보를 DB에 저장한다.<br>…5a2. 시스템은 초대장 내용을 기반으로 사용자의 캘린더에 일정을 등록한다.<br>…5a2. 사용자는 Use case #15에서 등록된 일정을 확인할 수 있다. |

**RELATED INFORMATION**

|                            |             |
|----------------------------|-------------|
| <b>Performance</b>         | ≤ 3 seconds |
| <b>Frequency</b>           | 알 수 없음      |
| <b>&lt;Concurrency&gt;</b> | 제한 없음       |
| <b>Due Date</b>            | 2023.11.30  |

**Use case #14 : Delete invitations list**
**GENERAL CHARACTERISTICS**

|                               |   |
|-------------------------------|---|
| <b>Summary</b>                | 사용자가 받은 초대장 목록을 전체 삭제한다.                |
| <b>Scope</b>                  | MeetMate                                |
| <b>Level</b>                  | User level                              |
| <b>Author</b>                 | 김수현                                     |
| <b>Last Update</b>            | 2023.11.18                              |
| <b>Status</b>                 | Analysis                                |
| <b>Primary Actor</b>          | User                                    |
| <b>Preconditions</b>          | 초대장 목록 보기 화면(Use case #13)이 열린 상태여야 한다. |
| <b>Trigger</b>                | 사용자가 초대장을 삭제하려고 할 때                     |
| <b>Success Post Condition</b> | 사용자는 초대장 목록을 전체 삭제할 수 있다.               |
| <b>Failed Post Condition</b>  | 사용자는 초대장 목록을 삭제할 수 없다.                  |

**MAIN SUCCESS SCENARIO**

| Step | Action                                  |
|------|---|
| S    | 사용자가 받은 초대장 목록을 삭제한다.                   |
| 1    | 사용자는 초대장 목록 보기 화면에 있는 삭제 아이콘을 누른다.      |
| 2    | 시스템은 초대장을 DB에서 삭제한 후 빈 초대장 목록 화면을 출력한다. |

**EXTENSION SCENARIOS**

|             |  |
|-------------|--|
| <b>Step</b> | Branching Action                                   |
| 2           | 2a. DB에서 삭제가 안 된 경우<br>...2a1. 초대장 목록 보기 화면이 출력된다. |

**RELATED INFORMATION**

|                            |             |
|----------------------------|-------------|
| <b>Performance</b>         | ≤ 3 seconds |
| <b>Frequency</b>           | 사용자당 평균 1번  |
| <b>&lt;Concurrency&gt;</b> | 제한 없음       |
| <b>Due Date</b>            | 2023.11.30  |

**Use case #15 : View calendar**
**GENERAL CHARACTERISTICS**

|                               |   |
|-------------------------------|---|
| <b>Summary</b>                | 사용자가 캘린더를 본다. 캘린더 화면에서 일정 목록과 일정 상세 정보를 볼 수 있다. |
| <b>Scope</b>                  | MeetMate  |
| <b>Level</b>                  | User level                                      |
| <b>Author</b>                 | 김수현   |
| <b>Last Update</b>            | 2023.11.18                                      |
| <b>Status</b>                 | Analysis  |
| <b>Primary Actor</b>          | User  |
| <b>Preconditions</b>          | 사용자가 calendar 탭을 눌러 캘린더 화면이 열린 상태여야 한다.         |
| <b>Trigger</b>                | 사용자가 캘린더 화면을 보려고 할 때                            |
| <b>Success Post Condition</b> | 사용자는 캘린더 화면에서 일정을 볼 수 있다.                       |
| <b>Failed Post Condition</b>  | 사용자는 캘린더 화면에서 일정을 볼 수 없다.                       |

**MAIN SUCCESS SCENARIO**

| Step | Action                                   |
|------|--|
| S    | 사용자는 캘린더 화면에서 캘린더 관련 기능을 사용하려 한다.        |
| 1    | 사용자는 캘린더 화면에서 생성된 일정들을 확인하고, 그중 하나를 누른다. |
| 2    | 시스템은 사용자가 누른 일정의 상세 정보를 출력한다.            |

**EXTENSION SCENARIOS**

| Step | Branching Action  |
|------|---|
| 1    | 1a. 일정 목록 확인<br>…1a1. 사용자는 캘린더 화면에서 생성된 일정들을 캘린더 UI로 모두 확인한다.<br>…1a2. 일정이 있으면, 캘린더 UI에 점 아이콘이 표시된다.<br>…1a3. 사용자는 좌·우 아이콘을 통해 과거, 현재, 미래의 일정을 모두 확인한다.<br>…1a4. 사용자가 캘린더 탭을 눌렀을 때 시스템은 현재 날짜를 기준으로 캘린더를 출력한다.<br>1b. 일정 확인<br>…1b1. 사용자는 캘린더 화면에서 원하는 일정을 직접 누른다.<br>…1b2. 시스템은 일정의 상세 정보를 캘린더 하단에 출력한다. (Use case #15-2)<br>…1b3. 시스템은 Use case #13에서 사용자가 수락한 초대장의 내용을 바탕으로 상세 정보를 출력한다.<br>1c. 등록된 일정이 없을 경우<br>…1c1. 시스템은 빈 캘린더 화면을 출력한다.<br>1d. 하루에 등록된 일정이 2개 이상일 경우<br>…1d1. 점 아이콘은 하나로 표시되고, 일정을 누르면 하단에 상세 정보로 모두 출력한다. |

| RELATED INFORMATION |               |
|---------------------|---------------|
| Performance         | ≤ 3 seconds   |
| Frequency           | 사용자당 하루 평균 2번 |
| <Concurrency>       | 제한 없음         |
| Due Date            | 2023.11.30    |

**Use case #16 : Manage my profile**
**GENERAL CHARACTERISTICS**

|                               |  |
|-------------------------------|--|
| <b>Summary</b>                | 사용자가 자신의 프로필을 관리한다. 시스템의 프로필 관련 기능(프로필 보기, 수정)을 설명한다.  |
| <b>Scope</b>                  | MeetMate   |
| <b>Level</b>                  | User level   |
| <b>Author</b>                 | 김수현  |
| <b>Last Update</b>            | 2023.11.01   |
| <b>Status</b>                 | Analysis   |
| <b>Primary Actor</b>          | User   |
| <b>Preconditions</b>          | 사용자는 로그인한 상태여야 한다.<br>Mypage 탭을 눌러 프로필 화면이 열린 상태여야 한다. |
| <b>Trigger</b>                | 사용자가 시스템을 이용해 사용자의 프로필을 관리하려고 할 때                      |
| <b>Success Post Condition</b> | 사용자는 시스템을 이용해서 프로필 관련 기능을 사용할 수 있다.                    |
| <b>Failed Post Condition</b>  | 사용자는 프로필 관련 기능을 사용할 수 없다.                              |

**MAIN SUCCESS SCENARIO**

| Step | Action                           |
|------|----------------------------------|
| S    | 사용자는 자신의 프로필을 관리하려 한다.           |
| 1    | 사용자는 프로필 화면에서 사용자의 프로필 정보를 확인한다. |
| 2    | 사용자는 프로필 화면에서 프로필 수정 아이콘을 누른다.   |
| 3    | 시스템은 프로필 입력 화면을 출력한다.            |
| 4    | 사용자는 프로필을 수정한 후 확인 버튼을 누른다.      |
| 5    | 시스템은 수정된 프로필 화면을 출력한다.           |
| 6    | 단계 1~7을 사용자가 원할 때까지 계속 반복한다.     |

**EXTENSION SCENARIOS**

| Step | Branching Action  |
|------|---|
| 1    | 1a. 내 프로필 보기<br>…1a1. 사용자가 Use case#1 Register에서 입력했던 정보(닉네임, 취미 카테고리, 위치)를 Mypage 화면에서 확인한다.<br>…1a2. 시스템은 사용자가 가입한 날과 현재 날짜의 차이를 계산하여 가입일을 Mypage 화면에 출력한다. |
| 4    | 4a. 내 프로필 수정<br>…4a1. 사용자는 프로필 화면에서 자신이 설정한 취미, 지역을 수정할 수 있다.   |

| RELATED INFORMATION |             |
|---------------------|-------------|
| Performance         | ≤ 3 seconds |
| Frequency           | 알 수 없음      |
| <Concurrency>       | 제한 없음       |
| Due Date            | 2023.11.30  |

**Use case #17 : Withdrawal**
**GENERAL CHARACTERISTICS**

|                               |  |
|-------------------------------|--|
| <b>Summary</b>                | 사용자가 시스템을 탈퇴한다.  |
| <b>Scope</b>                  | MeetMate   |
| <b>Level</b>                  | User level   |
| <b>Author</b>                 | 김수현  |
| <b>Last Update</b>            | 2023.11.01   |
| <b>Status</b>                 | Analysis   |
| <b>Primary Actor</b>          | User   |
| <b>Preconditions</b>          | 사용자는 로그인한 상태여야 한다.<br>Mypage 탭을 눌러 프로필 화면이 열린 상태여야 한다. |
| <b>Trigger</b>                | 로그인된 상태에서 Mypage 탭을 눌러 회원 탈퇴 버튼을 누를 때                  |
| <b>Success Post Condition</b> | 시스템에서 사용자 정보가 모두 지워지고, 사용자는 시스템 탈퇴에 성공한다.              |
| <b>Failed Post Condition</b>  | 사용자는 탈퇴에 실패해 로그인 상태를 유지한다.                             |

**MAIN SUCCESS SCENARIO**

| Step | Action                                |
|------|---------------------------------------|
| S    | 사용자가 회원 탈퇴를 한다.                       |
| 1    | 사용자는 MyPage 탭 프로필 화면에서 회원 탈퇴 버튼을 누른다. |
| 2    | 시스템은 사용자의 정보를 삭제한 후, 로그인 화면을 출력한다.    |
| 3    | 01 use case는 탈퇴에 성공하면 끝난다.            |

**EXTENSION SCENARIOS**

| Step | Branching Action  |
|------|---|
| 3    | 3a. 사용자가 시스템 탈퇴에 실패할 경우<br>...3a1. 시스템은 에러 메시지 출력한 후, MyPage 화면으로 돌아간다. |

**RELATED INFORMATION**

|                            |                  |
|----------------------------|------------------|
| <b>Performance</b>         | $\leq 5$ seconds |
| <b>Frequency</b>           | 사용자당 0~1번        |
| <b>&lt;Concurrency&gt;</b> | 제한 없음            |
| <b>Due Date</b>            | 2023.11.30       |

**Use case #18 : Withdrawal**
**GENERAL CHARACTERISTICS**

|                               |                                 |
|-------------------------------|---------------------------------|
| <b>Summary</b>                | 관리자는 사용자를 탈퇴시킨다.                |
| <b>Scope</b>                  | MeetMate                        |
| <b>Level</b>                  | User level                      |
| <b>Author</b>                 | 김수현                             |
| <b>Last Update</b>            | 2023.11.01                      |
| <b>Status</b>                 | Analysis                        |
| <b>Primary Actor</b>          | Admin                           |
| <b>Preconditions</b>          | 관리자는 MyPage 탭이 열려 있어야 한다.       |
| <b>Trigger</b>                | MyPage에서 탈퇴하기 버튼을 눌렀을 때         |
| <b>Success Post Condition</b> | 관리자가 탈퇴시키려는 사용자 정보가 시스템에서 삭제된다. |
| <b>Failed Post Condition</b>  | 관리자가 사용자를 탈퇴시키지 못한다.            |

**MAIN SUCCESS SCENARIO**

| <b>Step</b> | Action                               |
|-------------|--------------------------------------|
| S           | 관리자는 탈퇴하기 버튼을 누른다.                   |
| 1           | 시스템은 가입된 사용자들의 목록을 출력한다.             |
| 2           | 관리자는 탈퇴시키고자 하는 사용자를 누른다.             |
| 3           | 시스템은 탈퇴 확인 메시지를 출력한다.                |
| 4           | 관리자가 수락할 경우, 시스템은 사용자 정보를 DB에서 삭제한다. |

**EXTENSION SCENARIOS**

| <b>Step</b> | Branching Action   |
|-------------|--|
| 4           | 4a. 사용자를 삭제하는 데 실패하는 경우<br>…4a1. 시스템은 실패 메시지를 출력한다.<br>…4a2. 사용자 목록 화면으로 돌아간다. |

**RELATED INFORMATION**

|                            |                  |
|----------------------------|------------------|
| <b>Performance</b>         | $\leq 1$ seconds |
| <b>Frequency</b>           | 알 수 없음           |
| <b>&lt;Concurrency&gt;</b> | 제한 없음            |
| <b>Due Date</b>            | 2023.11.30       |

### 3. Class diagram

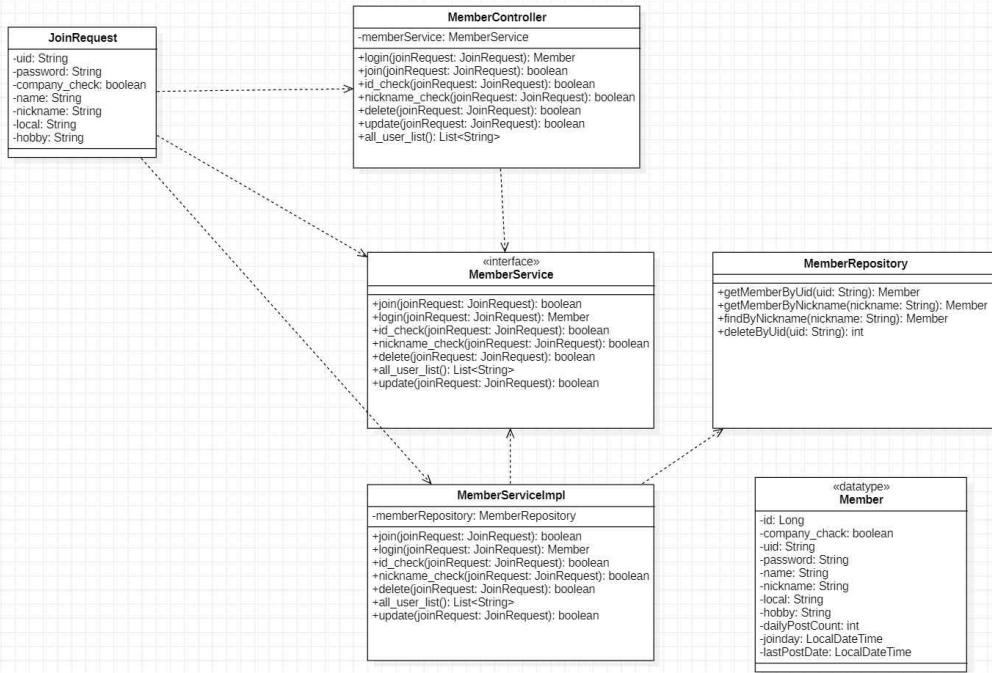
이 섹션에서는 'MeetMate'를 구조적인 관점에서 보고, 설계에 대한 제안을 제시한다. Use case analysis를 기반으로 식별된 사용자 관점의 기능들을 구현하는 데 필요한 객체들을 정의하고, 이런한 객체들 간의 관계, 속성, 메소드 등을 명시한다. Class Diagram은 하나 혹은 여러 개의 Use Case와 연결되어 있어 각 기능이 시스템의 특정 기능과 어떻게 연동되는지 확인할 수 있다.

'MeetMate'는 안드로이드 애플리케이션으로 동작하도록 설계되었으며, Class Diagram은 Server 및 Client의 관점으로 나누어져 있어 시스템의 구조를 명확하게 시각화한다. Description에서는 각 Class Diagram의 역할을 자세히 설명하며, 개체, 속성, 메소드가 시스템 내에서 어떠한 기능을 수행하는지에 대한 이해를 제공한다.

#### 3.1 Class diagram on Server Side

이 섹션에서는 Server의 관점에서 시스템의 구조를 파악하기 위한 Class Diagram을 작성한다. 코드의 구현은 스프링 부트를 활용하여 이루어졌으며, 전체 시스템은 내부적으로 세분화되어 있다. MVC 모델을 기반으로 하여 각 소단위 모델에 대한 클래스 디어그램이 작성되었다. 스프링 부트에서 기본적으로 제공되는 라이브러리에 대해서는 별도로 기술하지 않았다.

### 3.1.1 Member



| JoinRequest       |   |         |            |
|-------------------|---|---------|------------|
| Class description | Client로부터 받은 데이터를 담는 용도,<br>MemberController에서 사용되는 데이터 전송 객체를 정의 |         |            |
| 구분                | Name  | Type    | Visibility |
|                   | Description   |         |            |
| Attributes        | uid   | String  | private    |
|                   | 사용자의 아이디를 나타내는 속성   |         |            |
|                   | password  | String  | private    |
|                   | 사용자의 비밀번호를 나타내는 속성  |         |            |
|                   | company_check   | boolean | private    |
|                   | 기업 사용자 여부를 나타내는 속성,<br>true일 경우 기업 사용자를 나타냄                       |         |            |
|                   | name  | String  | private    |
|                   | 사용자의 실명을 나타내는 속성  |         |            |
|                   | nickname  | String  | private    |
|                   | 사용자의 닉네임을 나타내는 속성   |         |            |
|                   | local   | String  | private    |
|                   | 사용자의 위치 또는 주거지를 나타내는 속성   |         |            |
|                   | hobby   | String  | private    |
|                   | 사용자의 취미를 나타내는 속성  |         |            |

| <b>MemberController</b> |  |               |              |
|-------------------------|--|---------------|--------------|
| Class description       | Client로부터 들어온 요청을 관리   |               |              |
| 구분                      | Name   | Type          | Visibility   |
| Attributes              | memberService  | MemberService | private      |
|                         | MemberService 클래스의 인스턴스를 주입 받는 속성  |               |              |
| 구분                      | Name   | Argument      | Returns      |
|                         | Description  |               |              |
| Operations              | login  | JoinRequest   | Member       |
|                         | 로그인을 수행하는 메소드,<br>JoinRequest 객체를 받아 MemberService의 login 메소드를 호출하고, 그 결과로<br>Member 객체를 반환                            |               |              |
|                         | join   | JoinRequest   | boolean      |
|                         | 회원가입을 수행하는 메소드,<br>JoinRequest 객체를 받아 MemberService의 join 메소드를 호출하고, 그 결과로<br>회원가입 여부를 boolean 값으로 반환                  |               |              |
|                         | id_check   | JoinRequest   | boolean      |
|                         | 아이디 중복 확인을 수행하는 메소드,<br>JoinRequest 객체를 받아 MemberService의 id_check 메소드를 호출하고, 그 결과로<br>아이디 중복 여부를 boolean 값으로 반환       |               |              |
|                         | nickname_check   | JoinRequest   | boolean      |
|                         | 닉네임 중복 확인을 수행하는 메소드,<br>JoinRequest 객체를 받아 MemberService의 nickname_check 메소드를 호출하고, 그<br>결과로 닉네임 중복 여부를 boolean 값으로 반환 |               |              |
|                         | delete   | JoinRequest   | boolean      |
|                         | 회원 탈퇴를 수행하는 메소드,<br>JoinRequest 객체를 받아 MemberService의 delete 메소드를 호출하고, 그 결과로 회<br>원 탈퇴 여부를 boolean 값으로 반환             |               |              |
|                         | update   | JoinRequest   | boolean      |
|                         | 회원 정보 업데이트를 수행하는 메소드,<br>JoinRequest 객체를 받아 MemberService의 update 메소드를 호출하고, 그 결과로 회<br>원 정보 업데이트 여부를 boolean 값으로 반환   |               |              |
|                         | all_user_list  | none          | List<String> |
|                         | 모든 사용자의 닉네임을 리스트로 반환하는 메소드,<br>MemberService의 all_user_list 메소드를 호출하여 사용자 목록을 반환                                       |               |              |
|                         | send_profile   | JoinRequest   | Member       |
|                         | 사용자의 프로필 정보를 반환하는 메소드  |               |              |

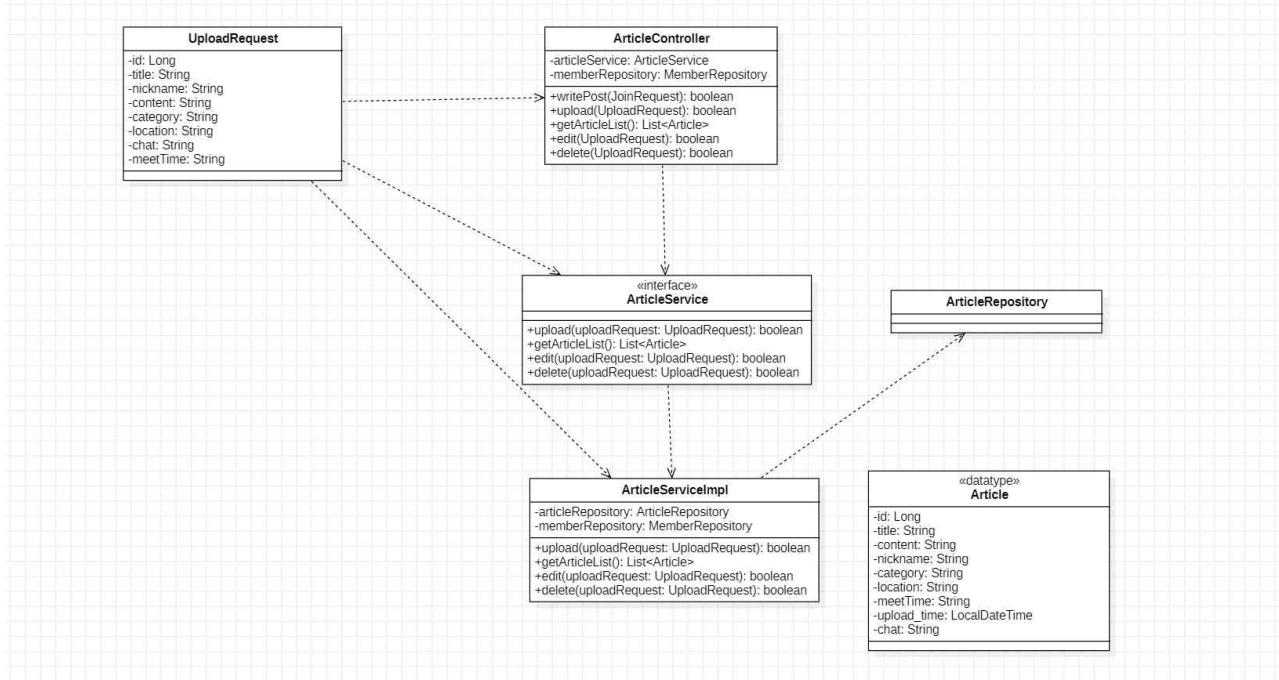
| <b>&lt;&lt;interface&gt;&gt; MemberService</b> |                                    |  |              |
|--|------------------------------------|--|--------------|
| Class description                              | Member Entity와 관련된 다양한 서비스 메소드를 선언 |  |              |
| 구분   | Name                               | Argument   | Returns      |
|  |                                    | Description  |              |
| Operations                                     | join                               | JoinRequest  | boolean      |
|  |                                    | 회원가입을 수행하는 메소드,<br>JoinRequest 객체를 매개변수로 받아 회원가입 여부를 반환                        |              |
|  | login                              | JoinRequest  | Member       |
|  |                                    | 로그인을 수행하는 메소드,<br>JoinRequest 객체를 매개변수로 받아 로그인한 회원의 정보를 담은<br>Member 객체를 반환    |              |
|  | id_check                           | JoinRequest  | boolean      |
|  |                                    | 아이디 중복 확인을 수행하는 메소드,<br>JoinRequest 객체를 매개변수로 받아 아이디 중복 여부를 반환                 |              |
|  | nickname_check                     | JoinRequest  | boolean      |
|  |                                    | 닉네임 중복 확인을 수행하는 메소드,<br>JoinRequest 객체를 매개변수로 받아 닉네임 중복 여부를 반환                 |              |
|  | delete                             | JoinRequest  | boolean      |
|  |                                    | 회원 탈퇴를 수행하는 메소드,<br>JoinRequest 객체를 매개변수로 받아 회원 탈퇴 여부를 반환                      |              |
|  | all_user_list                      |  | List<String> |
|  |                                    | 모든 사용자의 닉네임을 리스트로 반환하는 메소드   |              |
|  | update                             | JoinRequest  | boolean      |
|  |                                    | 회원정보 업데이트를 수행하는 메소드,<br>JoinRequest 객체를 매개변수로 받아 회원 정보 업데이트 여부를 반환             |              |
|  | send_profile                       | JoinRequest  | Member       |
|  |                                    | 사용자의 프로필 정보를 반환하는 메소드,<br>JoinRequest 객체를 매개변수로 받아 회원의 정보를 담은<br>Member 객체를 반환 |              |

| <b>MemberServiceImpl</b> |  |                  |              |
|--------------------------|--|------------------|--------------|
| Class description        | MemberService 인터페이스를 구현  |                  |              |
| 구분                       | Name   | Type             | Visibility   |
|                          | Description  |                  |              |
| Attributes               | memberRepository   | MemberRepository | private      |
|                          | Member Entity와 관련된 데이터베이스 작업 수행을 위한 인터페이스  |                  |              |
| 구분                       | Name   | Argument         | Returns      |
|                          | Description  |                  |              |
| Operations               | join   | JoinRequest      | boolean      |
|                          | 회원가입을 수행하는 메소드,<br>JoinRequest 객체에 필요한 정보를 추출하여 Member 객체를 생성,<br>생성된 Member를 데이터베이스에 저장하고 성공 유무를 반환   |                  |              |
|                          | login  | JoinRequest      | Member       |
|                          | 로그인을 수행하는 메소드,<br>JoinRequest 객체에서 uid와 password를 추출하여 해당하는 Member를 찾아<br>아로그인을 처리하고 결과로 Member 객체를 반환 |                  |              |
|                          | id_check   | JoinRequest      | boolean      |
|                          | 아이디 중복 확인을 수행하는 메소드,<br>JoinRequest 객체에서 uid를 추출하여 해당하는 Member를 찾아<br>중복 여부를 반환                        |                  |              |
|                          | nickname_check   | JoinRequest      | boolean      |
|                          | 닉네임 중복 확인을 수행하는 메소드,<br>JoinRequest 객체에서 nickname을 추출하여 해당하는 Member를 찾아<br>중복 여부를 반환                   |                  |              |
|                          | delete   | JoinRequest      | boolean      |
|                          | 회원 탈퇴를 수행하는 메소드,<br>JoinRequest 객체에서 uid를 추출하여 해당하는 Member를 삭제하고<br>삭제 성공 여부를 반환                       |                  |              |
|                          | all_user_list  |                  | List<String> |
|                          | 모든 사용자의 닉네임 목록을 반환하는 메소드,<br>모든 Member 객체를 조회,<br>각 Member의 uid를 리스트에 추가하여 반환                          |                  |              |
|                          | update   | JoinRequest      | boolean      |
|                          | 회원 정보 업데이트를 수행하는 메소드,<br>JoinRequest 객체에서 필요한 정보를 추출하여 해당하는 Member를 찾아<br>정보를 업데이트한 후, 업데이트 성공 유무를 반환  |                  |              |
|                          | send_profile   | JoinRequest      | Member       |
|                          | 사용자의 프로필 정보를 반환하는 메소드,<br>JoinRequest 객체를 매개변수로 받아 회원의 정보를 담은<br>Member 객체를 반환                         |                  |              |

| <b>&lt;&lt;interface&gt;&gt; MemberRepository</b> |  |          |         |
|---|--|----------|---------|
| Class description                                 | Member Entity와 관련된 데이터베이스 작업을 수행하는 메소드를 선언 |          |         |
| 구분  | Name                                       | Argument | Returns |
|   | Description                                |          |         |
| Operations  | getMemberByUid                             | String   | Member  |
|   | 주어진 아이디에 해당하는 Member 객체를 반환                |          |         |
|   | getMemberByNickname                        | String   | Member  |
|   | 주어진 닉네임에 해당하는 Member 객체를 반환                |          |         |
|   | findByNickname                             | String   | Member  |
|   | 주어진 닉네임에 해당하는 Member 객체를 반환                |          |         |
|   | deleteByUid                                | String   | int     |
| 주어진 아이디에 해당하는 Member 객체를 삭제                       |  |          |         |

| <b>Member</b>     |                                    |               |            |
|-------------------|------------------------------------|---------------|------------|
| Class description | Member Entity를 정의                  |               |            |
| 구분                | Name                               | Type          | Visibility |
|                   | Description                        |               |            |
| Attributes        | company_chack                      | boolean       | private    |
|                   | 회원이 기업 사용자인지 일반 사용자인지 나타내는 속성      |               |            |
| Attributes        | uid                                | String        | private    |
|                   | 회원의 아이디를 나타내는 속성                   |               |            |
| Attributes        | password                           | String        | private    |
|                   | 회원의 비밀번호를 나타내는 속성                  |               |            |
| Attributes        | name                               | String        | private    |
|                   | 회원의 실명을 나타내는 속성                    |               |            |
| Attributes        | nickname                           | String        | private    |
|                   | 회원의 닉네임을 나타내는 속성                   |               |            |
| Attributes        | local                              | String        | private    |
|                   | 회원의 위치나 주거지를 나타내는 속성               |               |            |
| Attributes        | hobby                              | String        | private    |
|                   | 회원의 취미를 나타내는 속성                    |               |            |
| Attributes        | dailyPostCount                     | int           | private    |
|                   | 회원의 일일 게시물 수를 나타내는 속성              |               |            |
| Attributes        | joinday                            | LocalDateTime | private    |
|                   | 회원이 가입한 날짜와 시간을 나타내는 속성            |               |            |
| Attributes        | lastPostDate                       | LocalDateTime | private    |
|                   | 회원이 마지막으로 게시물을 작성한 날짜와 시간을 나타내는 속성 |               |            |

### 3.1.2 Article



| UploadRequest     |                             |        |            |
|-------------------|-----------------------------|--------|------------|
| Class description | 게시물 업로드 요청을 처리하기 위한 객체      |        |            |
| 구분                | Name                        | Type   | Visibility |
|                   | Description                 |        |            |
| Attributes        | id                          | Long   | private    |
|                   | 게시물의 고유 식별자를 나타내는 속성        |        |            |
|                   | title                       | String | private    |
|                   | 게시물의 제목을 나타내는 속성            |        |            |
|                   | nickname                    | String | private    |
|                   | 게시물을 작성한 회원의 닉네임을 나타내는 속성   |        |            |
|                   | content                     | String | private    |
|                   | 게시물의 내용을 나타내는 속성            |        |            |
|                   | category                    | String | private    |
|                   | 게시물이 속한 카테고리를 나타내는 속성       |        |            |
| Associations      | location                    | String | private    |
|                   | 게시물이 관련된 위치나 주소 정보를 나타내는 속성 |        |            |
|                   | chat                        | String | private    |
|                   | 게시물과 관련된 채팅 정보를 나타내는 속성     |        |            |
| Relationships     | meetTime                    | String | private    |
|                   | 게시물에 명시된 모임 시간을 나타내는 속성     |        |            |

| <b>ArticleController</b>                     |   |                  |               |
|--|---|------------------|---------------|
| Class description                            | 게시물과 관련된 요청을 처리   |                  |               |
| 구분   | Name  | Type             | Visibility    |
|  | Description   |                  |               |
| Attributes                                   | articleService  | ArticleService   | private       |
|  | 게시물과 관련된 비즈니스 로직을 처리하는 ArticleService의 객체                     |                  |               |
|  | memberRepository  | MemberRepository | private       |
| 회원과 관련된 데이터베이스 작업을 처리하는 MemberRepository의 객체 |   |                  |               |
| 구분   | Name  | Argument         | Returns       |
|  | Description   |                  |               |
| Operations                                   | writePost   | JoinRequest      | boolean       |
|  | 일반 사용자 또는 기업사용자의 게시물 작성 여부를 판단                                |                  |               |
|  | upload  | UploadRequest    | boolean       |
|  | 게시물을 업로드하는 메소드,<br>UploadRequest 객체를 통해 게시물 작성에 필요한 정보를 전달 받음 |                  |               |
|  | getArticleList  | none             | List<Article> |
|  | 게시물 목록을 가져오는 메소드  |                  |               |
|  | edit  | UploadRequest    | boolean       |
|  | 게시물을 수정하는 메소드,<br>UploadRequest 객체를 통해 수정할 게시물의 정보를 전달 받음     |                  |               |
|  | delete  | UploadRequest    | boolean       |
|  | 게시물을 삭제하는 메소드,<br>UploadRequest 객체를 통해 삭제한 게시물의 정보를 전달 받음     |                  |               |

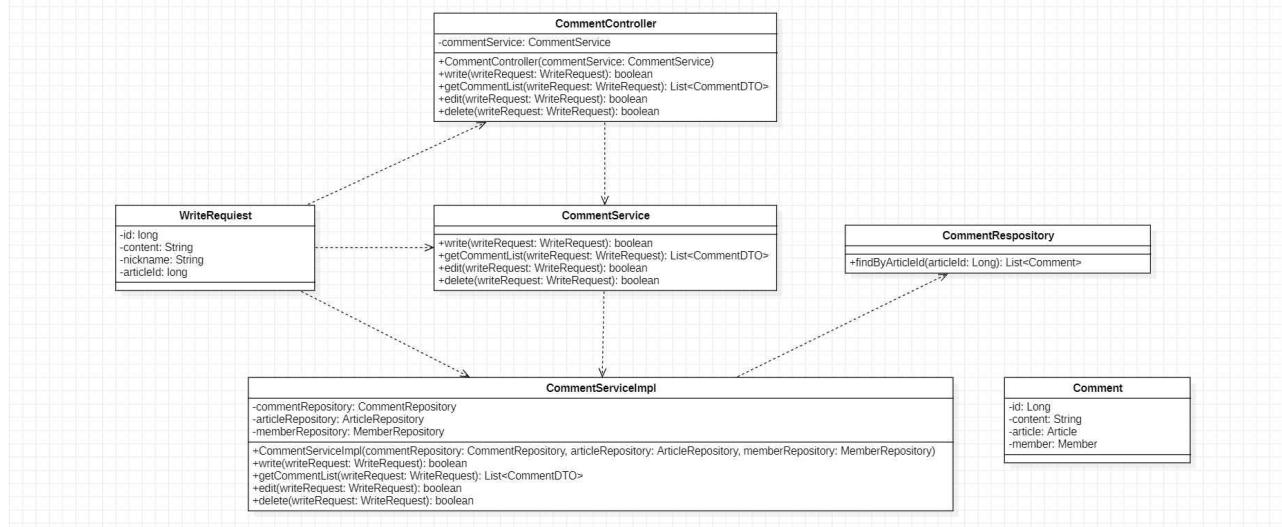
| <b>&lt;&lt;interface&gt;&gt; ArticleService</b> |                      |   |               |
|---|----------------------|---|---------------|
| Class description                               | 게시물과 관련된 비즈니스 로직을 정의 |   |               |
| 구분  | Name                 | Argument  | Returns       |
|   |                      | Description   |               |
| Operations                                      | upload               | UploadRequest   | boolean       |
|   |                      | 게시물을 업로드하는 메소드,<br>UploadRequest 객체를 받아 실제 업로드 작업을 수행하고,<br>게시물 업로드 성공 유무를 반환 |               |
|   | getArticleList       | none  | List<Article> |
|   |                      | 게시물 목록을 가져오는 메소드, Article 객체의 리스트를 반환   |               |
|   | edit                 | UploadRequest   | boolean       |
|   |                      | 게시물을 수정하는 메소드,<br>UploadRequest 객체를 받아 실제 수정 작업을 수행하고,<br>게시물 수정 성공 유무를 반환    |               |
| Operations                                      | delete               | UploadRequest   | boolean       |
|   |                      | 게시물을 삭제하는 메소드,<br>UploadRequest 객체를 받아 실제 삭제 작업을 수행하고,<br>게시물 삭제 유무를 반환       |               |
|   |                      |   |               |

| <b>ArticleServiceImpl</b> |                          |   |               |
|---------------------------|--------------------------|---|---------------|
| Class description         | ArticleService 인터페이스를 구현 |   |               |
| 구분                        | Name                     | Type  | Visibility    |
|                           |                          | Description   |               |
| Attributes                | articleRepository        | ArticleRepository   | private       |
|                           |                          | 게시물과 관련된 데이터베이스 작업을 처리  |               |
|                           | memberRepository         | MemberRepository  | private       |
|                           |                          | 회원과 관련된 데이터베이스 작업을 처리   |               |
|                           |                          |   |               |
|                           |                          |   |               |
| Operations                | Name                     | Argument  | Returns       |
|                           |                          | Description   |               |
|                           | upload                   | UploadRequest   | boolean       |
|                           |                          | 게시물을 업로드하는 메소드,<br>UploadRequest 객체를 통해 업로드에 필요한 정보를 받아오고,<br>게시물 업로드 성공 유무를 반환             |               |
|                           | getArticleList           | none  | List<Article> |
|                           |                          | 게시물 목록을 가져오는 메소드,<br>ArticleRepository에서 모든 게시물을 가져온 후,<br>각 게시물의 정보를 Article 객체로하여 리스트에 추가 |               |
| Operations                | edit                     | UploadRequest   | boolean       |
|                           |                          | 게시물을 수정하는 메소드,<br>UploadRequest 객체를 받아 수정에 필요한 정보를 받아오고,<br>게시물 수정 성공 유무를 반환                |               |
|                           | delete                   | UploadRequest   | boolean       |
|                           |                          | 게시물을 삭제하는 메소드,<br>UploadRequest 객체를 받아 삭제에 필요한 정보를 받아오고,<br>게시물 수정 성공 유무를 반환                |               |
|                           |                          |   |               |
|                           |                          |   |               |

| <b>&lt;&lt;interface&gt;&gt; ArticleRepository</b> |   |             |                   |
|--|---|-------------|-------------------|
| Class description                                  | Article Entity와 관련된 데이터베이스 작업을 수행하는 메소드를 선언                 |             |                   |
| Operations   | Name  | Argument    | Returns           |
|  |   | Description |                   |
|  | save  |             | Article           |
|  | 게시물을 저장 또는 수정하는 메소드,<br>Article 객체를 반환하여 저장 또는 수정된 게시물을 나타냄 |             |                   |
|  | findById  | Long        | <Optional>Article |
|  | 주어진 id에 해당하는 게시물을 조회하는 메소드                                  |             |                   |
|  | deleteById  | Long        | void              |
| 주어진 id에 해당하는 게시물을 삭제하는 메소드                         |   |             |                   |

| <b>Article</b>    |                        |               |            |
|-------------------|------------------------|---------------|------------|
| Class description | 게시물을 나타내는 Entity       |               |            |
| Attributes        | Name                   | Type          | Visibility |
|                   |                        | Description   |            |
|                   | id                     | long          | private    |
|                   | 게시물을 식별하기 위한 식별자 속성    |               |            |
|                   | title                  | String        | private    |
|                   | 게시물의 제목을 저장하는 속성       |               |            |
|                   | content                | String        | private    |
|                   | 게시물의 내용을 저장하는 속성       |               |            |
|                   | nickname               | String        | private    |
|                   | 게시물의 작성자 닉네임을 저장하는 속성  |               |            |
|                   | category               | String        | private    |
|                   | 게시물의 카테고리를 저장하는 속성     |               |            |
|                   | location               | String        | private    |
|                   | 게시물의 위치 정보를 저장하는 속성    |               |            |
|                   | meetTime               | String        | private    |
|                   | 게시물의 모임 시간을 저장하는 속성    |               |            |
|                   | upload_time            | LocalDateTime | private    |
|                   | 게시물이 업로드된 시간을 저장하는 속성  |               |            |
|                   | chat                   | String        | private    |
|                   | 게시물에 대한 채팅 내용을 저장하는 속성 |               |            |

### 3.1.3 Comment



| <b>WriteRequest</b> |   |        |            |
|---------------------|---|--------|------------|
| Class description   | 댓글 작성 요청을 나타냄                                       |        |            |
| 구분                  | Name  | Type   | Visibility |
|                     | Description   |        |            |
| Attributes          | id  | long   | private    |
|                     | 댓글을 식별하기 위한 속성                                      |        |            |
|                     | content   | String | private    |
|                     | 댓글의 내용을 나타내는 속성,<br>사용자가 작성한 댓글의 텍스트 내용이 저장         |        |            |
|                     | nickname  | String | private    |
|                     | 댓글 작성자의 닉네임을 나타내는 속성                                |        |            |
| Associations        | articleId   | long   | private    |
|                     | 댓글이 속한 게시물을 식별하기 위한 속성,<br>댓글이 특정 게시물에 연결되어 있음을 나타냄 |        |            |

| <b>CommentController</b> |   |                |                  |
|--------------------------|---|----------------|------------------|
| Class description        | 댓글과 관련된 Client의 요청을 처리                                      |                |                  |
| 구분                       | Name  | Type           | Visibility       |
|                          | Description   |                |                  |
| Attributes               | commentService  | CommentService | private          |
|                          | CommentService 클래스의 인스턴스를 주입 받는 속성                          |                |                  |
| 구분                       | Name  | Argument       | Returns          |
|                          | Description   |                |                  |
| Operations               | Write   | WriteRequest   | boolean          |
|                          | WriteRequest 객체에 담긴 정보를 통해 새로운 댓글을 작성하고,<br>댓글 작성 성공 유무를 반환 |                |                  |
|                          | getCommentList  | WriteRequest   | List<CommentDTO> |
|                          | WriteRequest 객체에 담긴 정보를 통해<br>해당 게시물에 속한 댓글 목록을 조회          |                |                  |
|                          | edit  | WriteRequest   | boolean          |
|                          | WriteRequest 객체에 담긴 정보를 통해<br>댓글을 수정하고 댓글 수정 성공 유무를 반환      |                |                  |
|                          | delete  | WriteRequest   | boolean          |
|                          | WriteRequest 객체에 담긴 정보를 통해<br>댓글을 삭제하고 댓글 삭제 성공 유무를 반환      |                |                  |

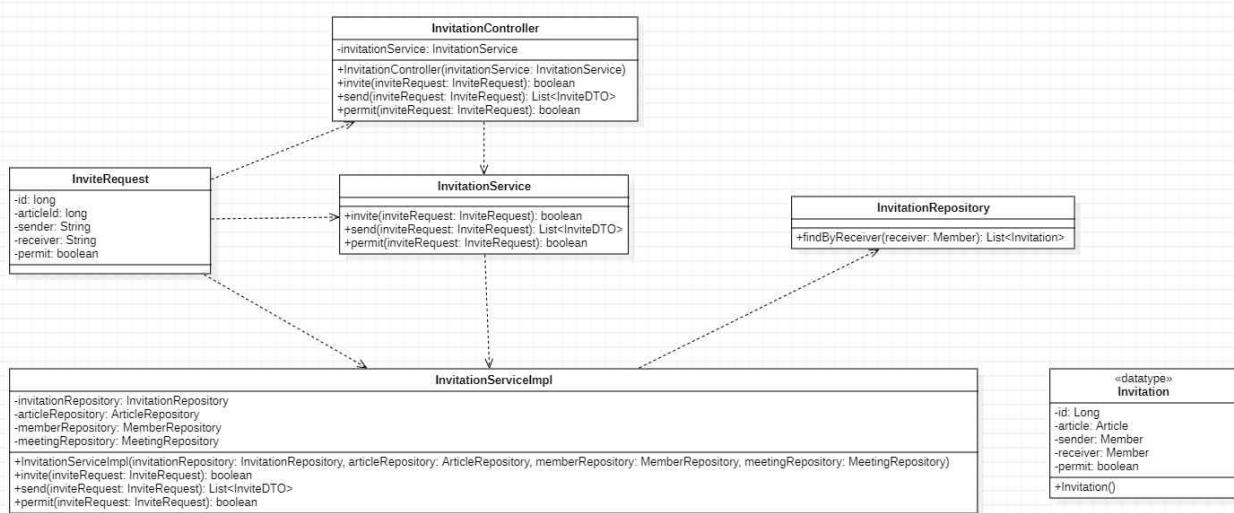
| <b>&lt;&lt;interface&gt;&gt; CommentService</b> |   |              |                  |
|---|---|--------------|------------------|
| Class description                               | 댓글과 관련된 서비스 기능을 정의한 인터페이스                                   |              |                  |
| 구분  | Name  | Argument     | Returns          |
|   | Description   |              |                  |
| Operations                                      | write   | WriteRequest | boolean          |
|   | WriteRequest 객체에 담긴 정보를 통해 새로운 댓글을 작성하고,<br>댓글 작성 성공 유무를 반환 |              |                  |
|   | getCommentList  | WriteRequest | List<CommentDTO> |
|   | WriteRequest 객체에 담긴 정보를 통해<br>해당 게시물에 속한 댓글 목록을 조회          |              |                  |
|   | edit  | WriteRequest | boolean          |
|   | WriteRequest 객체에 담긴 정보를 통해<br>댓글을 수정하고 댓글 수정 성공 유무를 반환      |              |                  |
|   | delete  | WriteRequest | boolean          |
|   | WriteRequest 객체에 담긴 정보를 통해<br>댓글을 삭제하고 댓글 삭제 성공 유무를 반환      |              |                  |

| <b>CommentServiceImpl</b>                                       |   |                   |                  |
|---|---|-------------------|------------------|
| Class description   | CommentService 인터페이스를 구현  |                   |                  |
| 구분  | Name  | Type              | Visibility       |
|   | Description   |                   |                  |
| Attributes  | commentRepository   | CommentRepository | private          |
|   | 댓글과 관련된 데이터베이스 조작을 수행   |                   |                  |
|   | articleRepository   | ArticleRepository | private          |
| 게시물과 관련된 데이터베이스 조작을 수행  |   |                   |                  |
| Operations  | memberRepository  | MemberRepository  | private          |
|   | 회원과 관련된 데이터베이스 조작을 수행   |                   |                  |
|   | write   | WriteRequest      | boolean          |
| WriteRequest 객체에 담긴 정보를 통해 특정 게시물에 댓글을 작성하고,<br>댓글 작성 성공 유무를 반환 |   |                   |                  |
| Operations  | getCommentList  | WriteRequest      | List<CommentDTO> |
|   | WriteRequest 객체에 담긴 정보를 통해<br>특정 게시물에 속한 댓글 목록을 조회하고,<br>조회된 댓글을 리스트 형태로 반환 |                   |                  |
|   | edit  | WriteRequest      | boolean          |
| WriteRequest 객체에 담긴 정보를 통해<br>댓글을 수정하고 댓글 수정 성공 유무를 반환          |   |                   |                  |
| Operations  | delete  | WriteRequest      | boolean          |
|   | WriteRequest 객체에 담긴 정보를 통해<br>댓글을 삭제하고 댓글 삭제 성공 유무를 반환                      |                   |                  |
|   |   |                   |                  |

| <<interface>> CommentRepository |   |          |               |
|---------------------------------|---|----------|---------------|
| Class description               | Comment Entity와 관련된 데이터베이스 작업을 수행하는 메소드를 선언 |          |               |
| 구분                              | Name  | Argument | Returns       |
|                                 | Description                                 |          |               |
| Operations                      | findByArticleId                             | long     | List<Comment> |
|                                 | 게시글의 ID를 이용하여 게시글에 속한 댓글의 리스트를 반환           |          |               |

| Comment           |                     |         |            |
|-------------------|---------------------|---------|------------|
| Class description | 댓글을 나타내는 Entity     |         |            |
| 구분                | Name                | Type    | Visibility |
|                   | Description         |         |            |
| Attributes        | id                  | long    | private    |
|                   | 댓글을 식별하기 위한 식별자 속성  |         |            |
|                   | content             | String  | private    |
|                   | 댓글의 내용을 나타내는 속성     |         |            |
|                   | article             | Article | private    |
|                   | 댓글이 속한 게시글을 나타내는 속성 |         |            |
|                   | member              | Member  | private    |
|                   | 댓글의 작성자를 나타내는 속성    |         |            |

### 3.1.4 Invitation



| InviteRequest                  |                           |        |            |
|--------------------------------|---------------------------|--------|------------|
| Class description              | Invitation에 관련된 요청을 나타냄   |        |            |
| 구분                             | Name                      | Type   | Visibility |
|                                | Description               |        |            |
| Attributes                     | id                        | long   | private    |
|                                | 초대장을 식별하기 위한 속성           |        |            |
|                                | articleId                 | long   | private    |
|                                | 초대장이 작성된 글을 식별하기 위한 속성    |        |            |
|                                | sender                    | String | private    |
|                                | 초대장을 보낸 사용자의 닉네임을 나타내는 속성 |        |            |
|                                | receiver                  | String | private    |
| 초대장을 받는 사용자의 닉네임을 나타내는 속성      |                           |        |            |
| permit                         |                           |        |            |
| 초대장에 반영된 모임에 대한 수락 여부를 나타내는 속성 |                           |        |            |

| InvitationController |  |                     |                 |
|----------------------|--|---------------------|-----------------|
| Class description    | 초대장에 대한 Client의 요청을 처리   |                     |                 |
| 구분                   | Name   | Type                | Visibility      |
|                      | Description  |                     |                 |
| Attributes           | invitationService  | InvitationService   | private         |
|                      | invitationService  | 클래스의 인스턴스를 주입 받는 속성 |                 |
| 구분                   | Name   | Argument            | Returns         |
|                      | Description  |                     |                 |
| Operations           | invite   | InviteRequest       | boolean         |
|                      | InviteRequest 객체에 담긴 정보를 통해, 초대장을 생성하고 초대장의 정보를 저장하고 초대장 생성 유무를 반환 |                     |                 |
|                      | send   | InviteRequest       | List<InviteDTO> |
|                      | InviteRequest 객체에 담긴 정보를 통해, 특정 사용자가 받은 초대장 목록을 반환                 |                     |                 |

|   |        |               |         |
|---|--------|---------------|---------|
|   | permit | InviteRequest | boolean |
| InviteRequest 객체에 담긴 정보를 통해,<br>받은 초대장의 초대를 수락을 처리하고<br>초대 수락 성공 유무를 반환 |        |               |         |

| <b>&lt;&lt;Interface&gt;&gt; InvitationService</b> |                        |  |               |                 |
|--|------------------------|--|---------------|-----------------|
| Class description                                  | 초대장과 관련된 기능을 정의한 인터페이스 |  |               |                 |
| Operations   | 구분                     | Name   | Argument      | Returns         |
|  |                        | invite   | InviteRequest | boolean         |
|  |                        | InviteRequest 객체에 담긴 정보를 통해,<br>초대장을 생성하고 초대장의 정보를 저장하고<br>초대장 생성 유무를 반환 |               |                 |
|  |                        | send   | InviteRequest | List<InviteDTO> |
|  |                        | InviteRequest 객체에 담긴 정보를 통해,<br>특정 사용자가 받은 초대장 목록을 반환                    |               |                 |
|  |                        | permit   | InviteRequest | boolean         |
|  |                        | InviteRequest 객체에 담긴 정보를 통해,<br>받은 초대장의 초대를 수락을 처리하고<br>초대 수락 성공 유무를 반환  |               |                 |

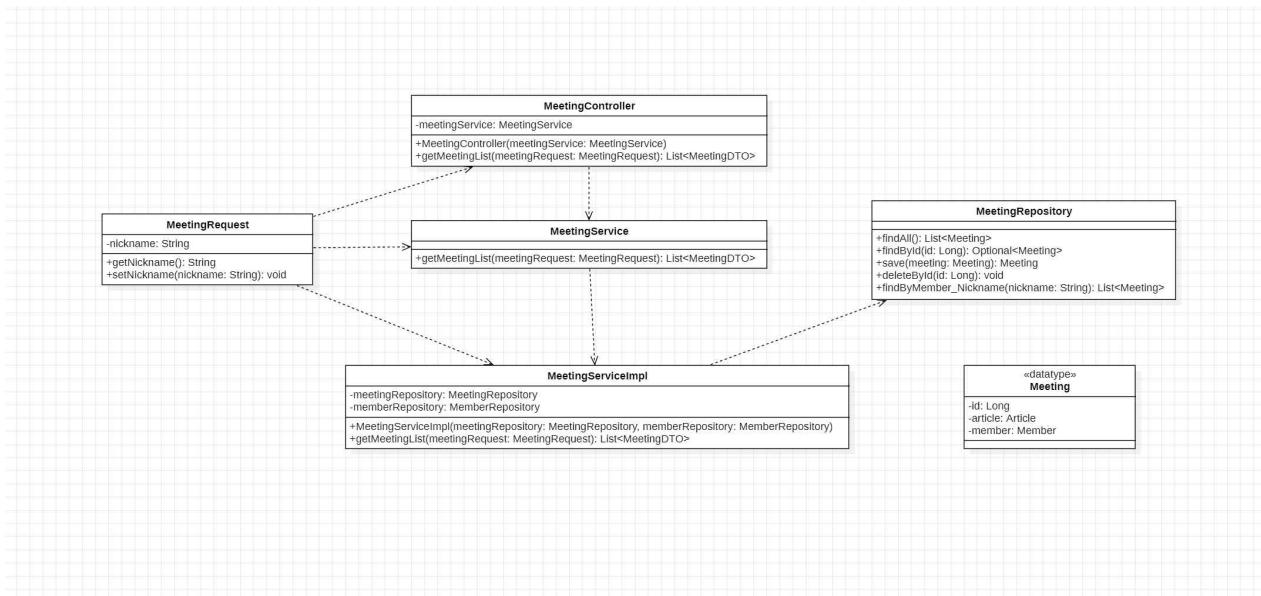
| <b>InvitationServiceImpl</b> |                             |  |                      |                 |
|------------------------------|-----------------------------|--|----------------------|-----------------|
| Class description            | InvitationService 인터페이스를 구현 |  |                      |                 |
| Attributes                   | 구분                          | Name   | Type                 | Visibility      |
|                              |                             | invitationRepository   | InvitationRepository | private         |
|                              |                             | 초대장과 관련된 데이터베이스 조작을 수행   |                      |                 |
|                              |                             | articleRepository  | ArticleRepository    | private         |
|                              |                             | 글과 관련된 데이터베이스 조작을 수행   |                      |                 |
|                              |                             | memberRepository   | MemberRepository     | private         |
| Operations                   | 구분                          | Name   | Argument             | Returns         |
|                              |                             | invite   | InviteRequest        | boolean         |
|                              |                             | InviteRequest 객체에 담긴 정보를 통해,<br>초대장을 생성하고 초대장의 정보를 저장하고<br>초대장 생성 유무를 반환 |                      |                 |
|                              |                             | send   | InviteRequest        | List<InviteDTO> |
|                              |                             | InviteRequest 객체에 담긴 정보를 통해,<br>특정 사용자가 받은 초대장 목록을 반환                    |                      |                 |
|                              |                             | permit   | InviteRequest        | boolean         |
|                              |                             | InviteRequest 객체에 담긴 정보를 통해,<br>받은 초대장의 초대를 수락을 처리하고<br>초대 수락 성공 유무를 반환  |                      |                 |

| <b>&lt;&lt;Interface&gt;&gt; InvitationRepository</b> |  |          |                  |
|---|--|----------|------------------|
| Class description                                     | Invitation Entity와 관련된 데이터베이스 작업을 수행하는 메소드를 선언 |          |                  |
| 구분  | Name   | Argument | Returns          |
|   | Description                                    |          |                  |
| Operations  | findByReceiver                                 | Member   | List<Invitation> |
|   | Member 객체의 닉네임을 통해 수신자의 초대장 목록을 조회하고 반환        |          |                  |

| <b>Invitation</b> |                            |         |            |
|-------------------|----------------------------|---------|------------|
| Class description | 초대장을 나타내는 Entity           |         |            |
| 구분                | Name                       | Type    | Visibility |
|                   | Description                |         |            |
| Attributes        | id                         | long    | private    |
|                   | 초대장을 식별하기 위한 속성            |         |            |
| Attributes        | articleId                  | Article | private    |
|                   | 초대장이 포함된 글을 식별하기 위한 속성     |         |            |
| Attributes        | sender                     | Member  | private    |
|                   | 글 작성자이자, 초대장 송신자를 나타내는 속성  |         |            |
| Attributes        | receiver                   | Member  | private    |
|                   | 댓글 작성자이자, 초대장 수신자를 나타내는 속성 |         |            |
| Attributes        | permit                     | boolean | private    |
|                   | 초대장의 수락 여부를 나타내는 속성        |         |            |

### 3.1.5 Meeting



| <b>Meeting</b>    |                            |          |            |
|-------------------|----------------------------|----------|------------|
| Class description | 모임을 나타내는 Entity 클래스        |          |            |
| 구분                | Name                       | Type     | Visibility |
|                   | Description                |          |            |
| Attributes        | id                         | Long     | private    |
|                   | 모임을 식별하기 위한 속성             |          |            |
|                   | article                    | Article  | private    |
|                   | 모임의 글과 관련된 데이터베이스 조작을 수행   |          |            |
| Operations        | member                     | Member   | private    |
|                   | 모임의 참가자와 관련된 데이터베이스 조작을 수행 |          |            |
| 구분                | Name                       | Argument | Returns    |
|                   | Description                |          |            |
| Operations        |                            |          |            |

| <b>MeetingRequest</b> |                          |        |            |
|-----------------------|--------------------------|--------|------------|
| Class description     | Meeting에 관련된 요청을 나타냄     |        |            |
| 구분                    | Name                     | Type   | Visibility |
|                       | Description              |        |            |
| Attributes            | nickname                 | String | private    |
|                       | 모임에 참가하는 사용자를 식별하기 위한 속성 |        |            |

| <b>MeetingController</b> |                       |                |            |
|--------------------------|-----------------------|----------------|------------|
| Class description        | 모임에 대한 Client의 요청을 처리 |                |            |
| 구분                       | Name                  | Type           | Visibility |
|                          | Description           |                |            |
| Attributes               | meetingService        | MeetingService | private    |

|            |   |                |                  |
|------------|---|----------------|------------------|
|            | meetingService 클래스의 인스턴스를 주입 받는 속성                  |                |                  |
| 구분         | Name  | Argument       | Returns          |
|            | Description   |                |                  |
| Operations | getMeetingList                                      | MeetingRequest | List<MeetingDTO> |
|            | MeetingRequest 객체의 정보를 통해<br>특정 사용자가 참가하는 모임 목록을 반환 |                |                  |

| <<interface>> MeetingService |   |                |                  |
|------------------------------|---|----------------|------------------|
| Class description            | 모임과 관련된 기능을 제공하는 인터페이스                              |                |                  |
| 구분                           | Name  | Argument       | Returns          |
|                              | Description   |                |                  |
| Operations                   | getMeetingList                                      | MeetingRequest | List<MeetingDTO> |
|                              | MeetingRequest 객체의 정보를 통해<br>특정 사용자가 참가하는 모임 목록을 반환 |                |                  |

| <<interface>> MeetingRepository |  |          |               |
|---------------------------------|--|----------|---------------|
| Class description               | Meeting Entity와 관련된 데이터베이스 작업을 수행하는 메소드를 선언        |          |               |
| 구분                              | Name   | Argument | Returns       |
|                                 | Description  |          |               |
| Operations                      | findByMember_Nickname                              | String   | List<Meeting> |
|                                 | 사용자의 닉네임을 기반으로 사용자가 참가하는 모든 모임을 조회하고,<br>모임 목록을 반환 |          |               |

| MeetingServicImpl |   |                   |                  |
|-------------------|---|-------------------|------------------|
| Class description | MeetingService 인터페이스를 구현                            |                   |                  |
| 구분                | Name  | Type              | Visibility       |
|                   | Description   |                   |                  |
| Attributes        | meetingRepository                                   | MeetingRepository | private          |
|                   | 모임과 관련된 데이터베이스 조작을 수행                               |                   |                  |
| Operations        | memberRepository                                    | MemberRepository  | private          |
|                   | 회원과 관련된 데이터베이스 조작을 수행                               |                   |                  |
| 구분                | Name  | Argument          | Returns          |
|                   | Description   |                   |                  |
| Operations        | getMeetingList                                      | MeetingRequest    | List<MeetingDTO> |
|                   | MeetingRequest 객체의 정보를 통해<br>특정 사용자가 참가하는 모임 목록을 반환 |                   |                  |

### 3.2 Class diagram on Client Side

이 섹션에서는 Client의 관점에서 시스템의 구조를 파악하기 위한 Class Diagram을 작성한다. 코드의 구현은 Flutter 프레임워크를 활용하여 이루어졌으며, 전체 시스템을 다음의 3가지로 분류하여 클래스 다이어그램을 구성하였다. UI 컴포넌트는 Class Diagram에서 제외되었으며, 사용된 기본 컴포넌트에 대한 부가 설명은 기술하지 않았다.

#### 1. Topappbar

애플리케이션의 상단에 위치하여 사용자가 주요 기능에 빠르게 접근할 수 있는 공간으로, “페이지에 대한 설명”, “뒤로가기 버튼”, “초대장 버튼”, “검색 버튼”을 포함하고 있다.

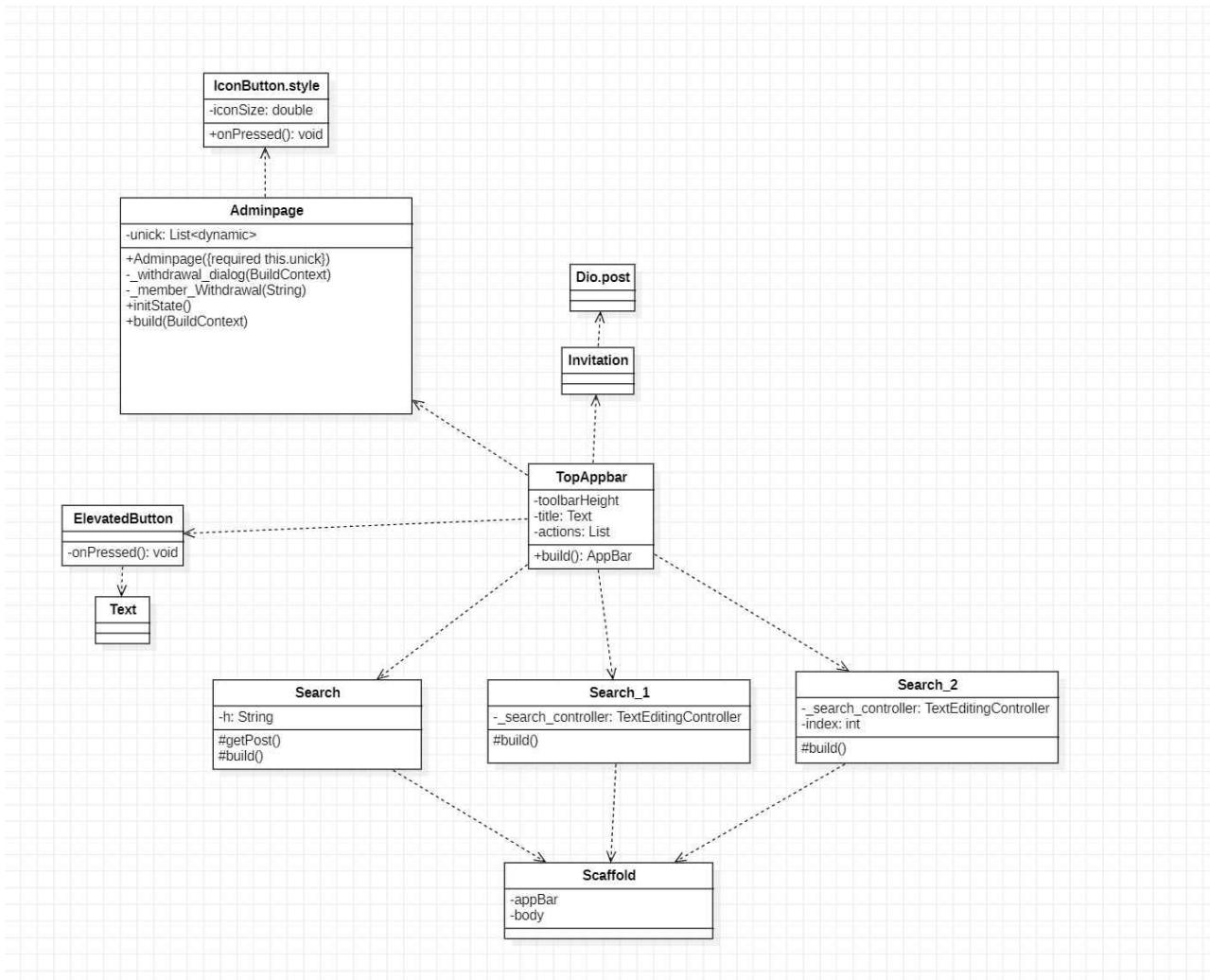
#### 2. Bottomappbar

애플리케이션의 하단에 위치하여 사용자에게 글 목록 출력을 위한 여러 옵션을 제공한다.

#### 3. Body

Topappbar와 Bottomappbar를 제외한 UI의 중간 부분으로, 애플리케이션의 주요 콘텐츠를 담고 있다.

### 3.2.1 Topappbar



| Adminpage         |   |                       |            |
|-------------------|---|-----------------------|------------|
| Class description | 'adminpage'라는 StatelessWidget 클래스를 정의 클래스 |                       |            |
| 구분                | Name                                      | Type                  | Visibility |
|                   | Description                               |                       |            |
| Attributes        | unlck                                     | List<dynamic>         | private    |
|                   | 동적인 객체들의 리스트                              |                       |            |
| 구분                | Name                                      | Argument              | Returns    |
|                   | Description                               |                       |            |
| Operations        | Adminpage                                 | {required this.unlck} | none       |
|                   | 'unlck'이라는 필수 매개변수를 받음                    |                       |            |
|                   | _withdrawal_dialog                        | BuildContext          | none       |
|                   | 성공 메시지와 함께 대화상자를 표시                       |                       |            |
|                   | _member_Withdrawal                        | String                | none       |
|                   | 회원 탈퇴 작업을 수행                              |                       |            |
|                   | InitState                                 | none                  | none       |
|                   | 위젯 라이프사이클의 일부로, 일회성 초기화 작업                |                       |            |
|                   | build                                     | BuildContext          | none       |
| 위젯의 구조를 정의        |   |                       |            |

| TopAppbar            |   |          |            |
|----------------------|---|----------|------------|
| Class<br>description | 상단에 위치한 AppBar를 정의 클래스  |          |            |
| 구분                   | Name  | Type     | Visibility |
|                      | Description   |          |            |
| Attributes           | toolbarHeight   | none     | private    |
|                      | AppBar의 높이를 설정  |          |            |
|                      | title   | Text     | private    |
|                      | AppBar에 표시되는 텍스트로 사용자의 지역 정보를 표시  |          |            |
| Operations           | actions   | List     | private    |
|                      | AppBar에 추가되는 다양한 액션이 들어감  |          |            |
|                      | Name  | Argument | Returns    |
|                      | Description   |          |            |
| 구분                   | build   | none     | Appbar     |
|                      | AppBar 생성   |          |            |
| ElevatedButton       |   |          |            |
| Class<br>description | 사용자 닉네임이 'admin'인 경우 Adminpage로 이동하는 버튼 클래스                               |          |            |
| Text                 |   |          |            |
| Class<br>description | Text 출력 클래스   |          |            |
| IconButton           |   |          |            |
| Class<br>description | 검색 버튼 및 초대장 버튼 클래스  |          |            |
| Invitation           |   |          |            |
| Class<br>description | StatefulWidget인 invitation클래스를 정의하고, 상태를 관리하는 _InvitationState클래스를 반환 클래스 |          |            |
| Dio.post             |   |          |            |
| Class<br>description | Invitation 페이지에서 초대장을 수락 또는 거절하는 클래스                                      |          |            |
| IconButton.style     |   |          |            |
| Class<br>description | IconButton의 스타일을 설정하는 클래스   |          |            |
| 구분                   | Name  | Type     | Visibility |
|                      | Description   |          |            |
| Attributes           | iconSize  | double   | private    |
|                      | IconButton 크기 설정  |          |            |
| Operations           | Name  | Argument | Returns    |
|                      | Description   |          |            |
| Operations           | onPressed   | none     | void       |
|                      | IconButton이 눌렸을 시 설정  |          |            |

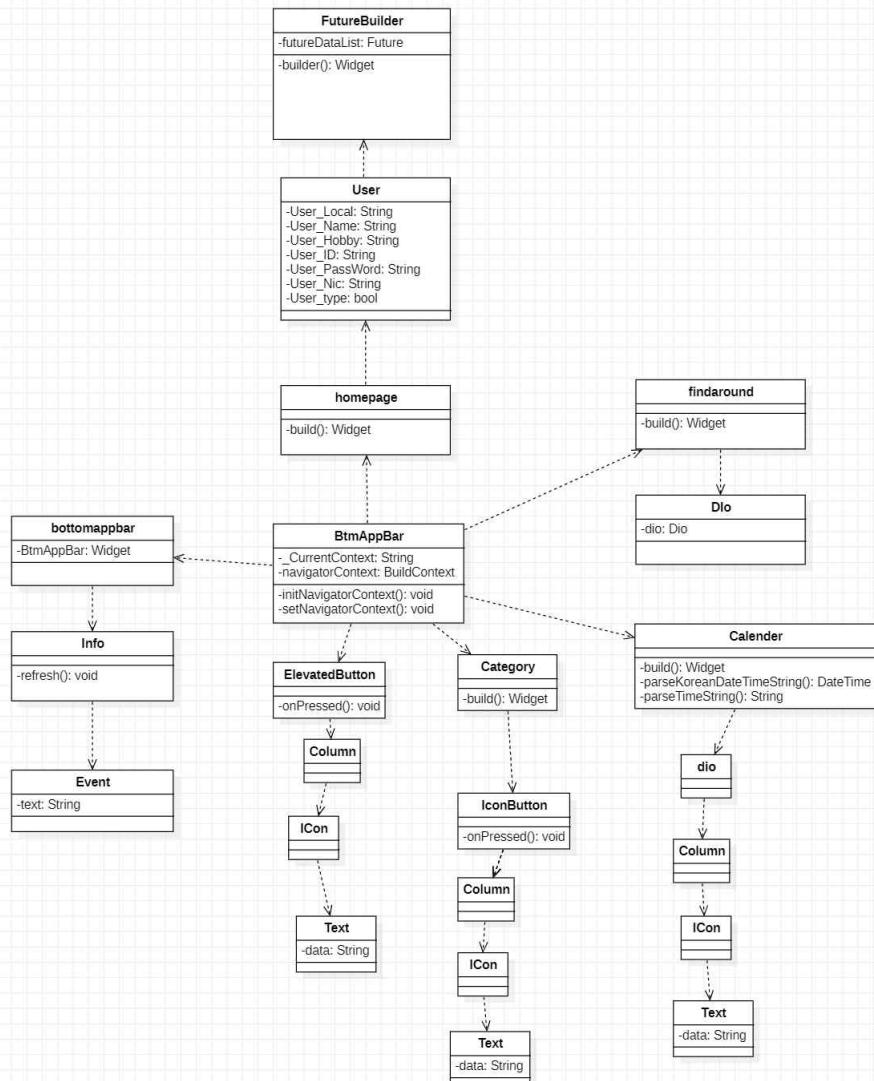
| Search            |  |          |            |
|-------------------|--|----------|------------|
| Class description | 'StatefulWidget'을 상속하며, 검색화면을 나타내는 클래스                 |          |            |
| 구분                | Name   | Type     | Visibility |
|                   | Description  |          |            |
| Attributes        | h  | String   | private    |
|                   | 검색할 카테고리를 가져옴  |          |            |
| 구분                | Name   | Argument | Returns    |
|                   | Description  |          |            |
| Operations        | getPost  | none     | none       |
|                   | Dio를 사용하여 서버로부터 데이터를 가져오는 비동기 메서드                      |          |            |
| Operations        | build  | none     | none       |
|                   | 데이터를 성공적으로 가져오면 'ListView.builder'를 사용하여 검색 결과를 화면에 출력 |          |            |

| Search_1          |  |                       |            |
|-------------------|--|-----------------------|------------|
| Class description | 'StatefulWidget'을 상속하며, 취미 목록을 그리드 형태로 표시하는 화면을 나타내는 클래스 |                       |            |
| 구분                | Name   | Type                  | Visibility |
|                   | Description  |                       |            |
| Attributes        | _search_controller                                       | TextEditingController | private    |
|                   | 검색 창에 입력된 텍스트를 제어하는 데 사용                                 |                       |            |
| 구분                | Name   | Argument              | Returns    |
|                   | Description  |                       |            |
| Operations        | build  | none                  | none       |
|                   | 'AppBar'을 사용하여 상단에 타이틀을 표시                               |                       |            |

| Search_2          |   |                       |            |
|-------------------|---|-----------------------|------------|
| Class description | 'StatefulWidget'을 상속하며, 특정 취미에 대한 세부 정보를 표시하고 그에 속하는 하위 취미를 선택할 수 있는 화면을 나타내는 클래스 |                       |            |
| 구분                | Name  | Type                  | Visibility |
|                   | Description   |                       |            |
| Attributes        | _search_controller  | TextEditingController | private    |
|                   | 검색 창에 입력된 텍스트를 제어하는 데 사용  |                       |            |
| Attributes        | index   | int                   | private    |
|                   | 인덱스를 설정   |                       |            |
| 구분                | Name  | Argument              | Returns    |
|                   | Description   |                       |            |
| Operations        | build   | none                  | none       |
|                   | 'AppBar'을 사용하여 상단에 타이틀을 표시  |                       |            |

| Scaffold          |                          |      |            |
|-------------------|--------------------------|------|------------|
| Class description | 기본적인 디자인 구조를 제공하는 위젯 클래스 |      |            |
| 구분                | Name                     | Type | Visibility |
|                   | Description              |      |            |
| Attributes        | appBar                   | none | private    |
|                   | 앱 상단에 표시되는 AppBar를 정의    |      |            |
| Attributes        | body                     | none | private    |
|                   | 앱 화면의 주요 콘텐츠를 정의         |      |            |

### 3.2.2 bottomappbar



| BtmAppBar  |                                 |              |            |
|--|---------------------------------|--------------|------------|
| Class description                                | 화면 하단에 위치하는 바텀 앱 바를 정의 클래스      |              |            |
| 구분   | Name                            | Type         | Visibility |
|  | Description                     |              |            |
| Attributes                                       | _CurrentContext                 | String       | private    |
|  | 현재 화면의 상태를 나타내는 문자열             |              |            |
| Operations                                       | navigatorContext                | BuildContext | private    |
|  | BuildContext를 저장하는 변수           |              |            |
| 구분   | Name                            | Argument     | Returns    |
|  | Description                     |              |            |
| Operations                                       | initNavigatorContext            | none         | void       |
|  | 메소드는 초기에 'navigatorContext'를 설정 |              |            |
|  | setNavigatorContext             | none         | void       |
| 현재 context와 저장된 값을 비교하여 'navigatorContext'를 업데이트 |                                 |              |            |

| ElevatedButton    |                       |          |         |
|-------------------|-----------------------|----------|---------|
| Class description | 머티리얼 디자인의 일반적인 버튼 클래스 |          |         |
| 구분                | Name                  | Argument | Returns |
|                   | Description           |          |         |
| Operations        | onPressed             | none     | void    |
|                   | 버튼을 눌렀을 때 설정          |          |         |

| Column            |              |
|-------------------|--------------|
| Class description | 버튼 개수 정의 클래스 |

| ICon              |               |
|-------------------|---------------|
| Class description | 버튼 아이콘 정의 클래스 |

| Text              |              |        |            |
|-------------------|--------------|--------|------------|
| Class description | 버튼 문자 정의 클래스 |        |            |
| 구분                | Name         | Type   | Visibility |
|                   | Description  |        |            |
| Attributes        | data         | String | private    |
|                   | 버튼 문자열 정보    |        |            |

| Category          |                                  |          |         |
|-------------------|----------------------------------|----------|---------|
| Class description | 화면 하단에 위치하는 바텀 앱 바의 카테고리를 정의 클래스 |          |         |
| 구분                | Name                             | Argument | Returns |
|                   | Description                      |          |         |
| Operations        | build                            | none     | Widget  |
|                   | 카테고리 버튼 생성                       |          |         |

| IconButton        |                     |          |         |
|-------------------|---------------------|----------|---------|
| Class description | 카테고리 버튼의 아이콘 정의 클래스 |          |         |
| 구분                | Name                | Argument | Returns |
|                   | Description         |          |         |
| Operations        | onPressed           | none     | void    |
|                   | 버튼을 눌렀을 때 설정        |          |         |

| Column            |                |
|-------------------|----------------|
| Class description | 버튼의 인덱스 정보 클래스 |

| ICon              |                    |
|-------------------|--------------------|
| Class description | 카테고리 버튼 아이콘 정의 클래스 |

| Text              |                   |        |            |
|-------------------|-------------------|--------|------------|
| Class description | 카테고리 버튼 문자 정의 클래스 |        |            |
| 구분                | Name              | Type   | Visibility |
|                   | Description       |        |            |
| Attributes        | data              | String | private    |
|                   | 버튼 문자열 정보         |        |            |

| Calender          |                                |          |         |
|-------------------|--------------------------------|----------|---------|
| Class description | 화면 하단에 위치하는 바텀 앱 바의 캘린더 정의 클래스 |          |         |
| 구분                | Name                           | Argument | Returns |
|                   | Description                    |          |         |
| Operations        | build                          | none     | Widget  |
|                   | 캘린더 버튼 생성                      |          |         |

| dio               |               |  |  |
|-------------------|---------------|--|--|
| Class description | 비동기 작업 요청 클래스 |  |  |

| Column            |                |  |  |
|-------------------|----------------|--|--|
| Class description | 버튼의 인덱스 정보 클래스 |  |  |

| Icon              |                   |  |  |
|-------------------|-------------------|--|--|
| Class description | 캘린더 버튼 아이콘 정의 클래스 |  |  |

| Text              |                   |        |            |
|-------------------|-------------------|--------|------------|
| Class description | 캘린더 버튼 문자열 정의 클래스 |        |            |
| 구분                | Name              | Type   | Visibility |
|                   | Description       |        |            |
| Attributes        | data              | String | private    |
|                   | 문자열 정보            |        |            |

| bottomappbar      |                           |        |            |
|-------------------|---------------------------|--------|------------|
| Class description | 앱 하단에 위치하는 바텀 앱 바를 정의 클래스 |        |            |
| 구분                | Name                      | Type   | Visibility |
|                   | Description               |        |            |
| Attributes        | BtmAppBar                 | Widget | private    |
|                   | 'bottomappbar'의 위치        |        |            |

| Info              |                         |          |         |
|-------------------|-------------------------|----------|---------|
| Class description | 화면 하단에 위치하는 내 정보 정의 클래스 |          |         |
| 구분                | Name                    | Argument | Returns |
|                   | Description             |          |         |
| Operations        | refresh                 | none     | void    |
|                   | 내 정보 새로고침 기능            |          |         |

| Event             |                  |        |            |
|-------------------|------------------|--------|------------|
| Class description | 상태의 변경을 나타내는 클래스 |        |            |
| 구분                | Name             | Type   | Visibility |
|                   | Description      |        |            |
| Attributes        | text             | String | private    |
|                   | 변경에 대한 문자열 정보    |        |            |

| homepage          |                               |          |         |
|-------------------|-------------------------------|----------|---------|
| Class description | 화면 하단에 위치하는 버튼 앱 바의 홈을 정의 클래스 |          |         |
| 구분                | Name                          | Argument | Returns |
|                   | Description                   |          |         |
| Operations        | build                         | none     | Widget  |
|                   | 홈 버튼 생성                       |          |         |

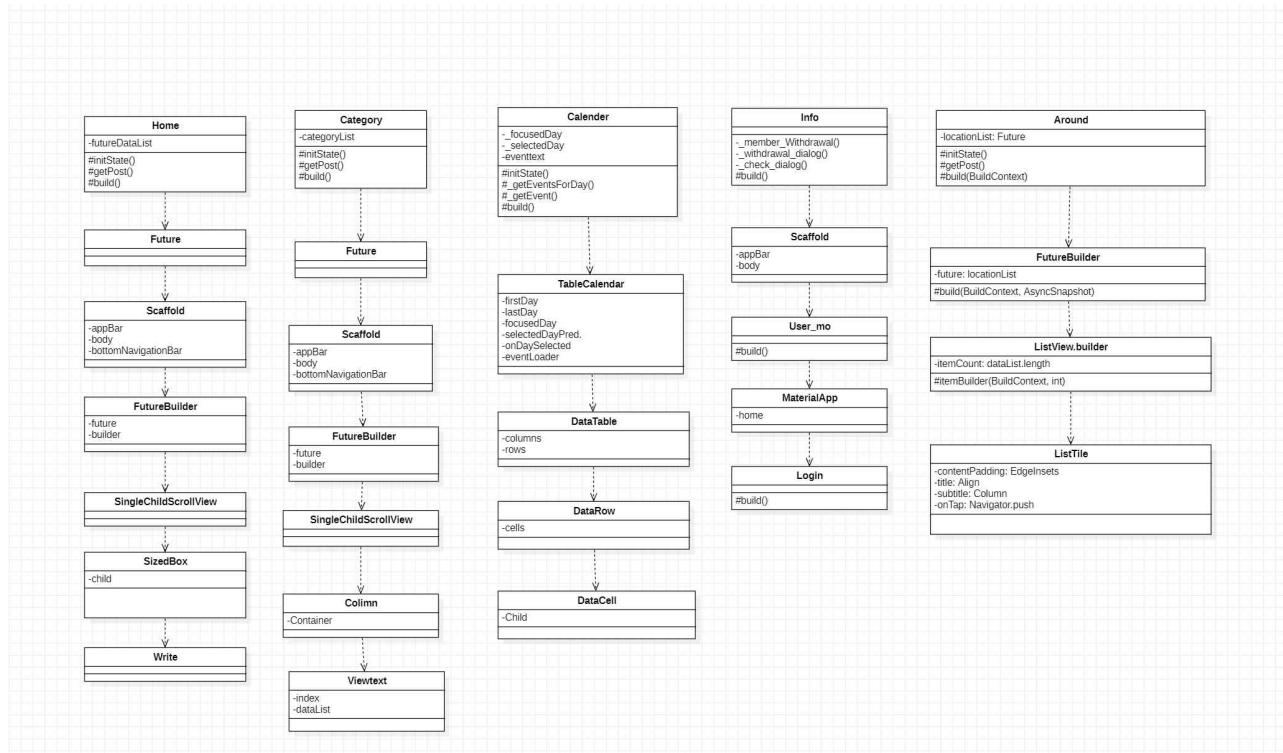
| User              |               |        |            |
|-------------------|---------------|--------|------------|
| Class description | 사용자 정보 클래스    |        |            |
| 구분                | Name          | Type   | Visibility |
|                   | Description   |        |            |
| Attributes        | User_Local    | String | private    |
|                   | 사용자 지역        |        |            |
|                   | User_Name     | String | private    |
|                   | 사용자 이름        |        |            |
|                   | User_Hobby    | String | private    |
|                   | 사용자 취미        |        |            |
|                   | User_ID       | String | private    |
|                   | 사용자 아이디       |        |            |
|                   | User_PassWord | String | private    |
|                   | 사용자 비밀번호      |        |            |
| Operations        | User_Nic      | String | private    |
|                   | 사용자 닉네임       |        |            |
| Attributes        | User_type     | bool   | private    |
|                   | 사용자 분류(기업/개인) |        |            |

| FutureBuilder     |                               |        |            |
|-------------------|-------------------------------|--------|------------|
| Class description | 비동기적인 연산 결과를 기반으로 UI 간신하는 클래스 |        |            |
| 구분                | Name                          | Type   | Visibility |
|                   | Description                   |        |            |
| Attributes        | futureDataList                | Future | private    |
|                   | 비동기 작업의 결과를 담기 위한 변수로 위젯에서 사용 |        |            |
| Operations        | builder                       | none   | Widget     |
|                   | UI 간신하여 생성                    |        |            |

| findaround        |                                  |          |         |
|-------------------|----------------------------------|----------|---------|
| Class description | 화면 하단에 위치하는 바텀 앱 바의 주변취미를 정의 클래스 |          |         |
| 구분                | Name                             | Argument | Returns |
|                   | Description                      |          |         |
| Operations        | build                            | none     | Widget  |
|                   | 주변취미 버튼을 생성                      |          |         |

| DIO               |               |      |            |
|-------------------|---------------|------|------------|
| Class description | 비동기 작업 요청 클래스 |      |            |
| 구분                | Name          | Type | Visibility |
|                   | Description   |      |            |
| Attributes        | dio           | Dio  | private    |
|                   | 비동기 작업 요청 변수  |      |            |

### 3.2.3 body



| Calender          |  |          |  |
|-------------------|--|----------|--|
| Class description | 캘린더 위젯을 정의하는 클래스                                     |          |  |
| 구분                | Name   | Type     | Visibility                                 |
|                   |  |          | Description                                |
| Attributes        | _focusedDay  | none     | private                                    |
|                   | 현재 화면에 표시된 달력에서 포커스된 날짜를 나타내는 속성                     |          |  |
|                   | _selectedDay   | none     | private                                    |
|                   | 사용자가 선택한 날짜를 나타내는 속성                                 |          |  |
|                   | eventtext  | none     | private                                    |
|                   |  |          | 선택된 날짜에 해당하는 이벤트 텍스트를 담는 'List<String>'    |
| 구분                | Name   | Argument | Returns                                    |
|                   |  |          | Description                                |
| Operations        | initState  | none     | none                                       |
|                   | 상태를 초기화하는 메소드  |          |  |
|                   | _getEventsForDay                                     | none     | none                                       |
|                   | 주어진 날짜에 대한 이벤트 목록을 반환하는 메소드                          |          |  |
|                   | _getEvent  | none     | none                                       |
|                   | '_focusedDay'에 해당하는 이벤트를 가져와 'eventtext'를 업데이트하는 메소드 |          |  |
|                   |  |          | 달력과 이벤트 목록을 표시하고, 새로고침 버튼이 있는 'AppBar'을 구성 |

| TableCalendar                |  |      |            |  |  |  |
|------------------------------|--|------|------------|--|--|--|
| Class description            | 달력을 표시하는 위젯으로, 선택한 날짜에 대한 이벤트를 표시하고 업데이트하는 클래스 |      |            |  |  |  |
| 구분                           | Name   | Type | Visibility |  |  |  |
|                              | Description                                    |      |            |  |  |  |
| Attributes                   | firstDay                                       | none | private    |  |  |  |
|                              | 첫 번째 날짜의 정보를 가진 속성                             |      |            |  |  |  |
|                              | lastDay  | none | private    |  |  |  |
|                              | 마지막 날짜의 정보를 가진 속성                              |      |            |  |  |  |
|                              | focusedDay                                     | none | private    |  |  |  |
|                              | 현재 캘린더 위젯이 초점을 맞춘 날짜를 나타내는 변수                  |      |            |  |  |  |
|                              | selectedDayPred                                | none | private    |  |  |  |
|                              | 이전에 선택된 날짜를 나타내는 속성                            |      |            |  |  |  |
|                              | onDaySelected                                  | none | private    |  |  |  |
|                              | 선택된 날짜를 나타내는 속성                                |      |            |  |  |  |
| eventLoader                  |  |      |            |  |  |  |
| 각 날짜의 이벤트를 동적으로 로드하기 위한 콜백함수 |  |      |            |  |  |  |

| DataTable                          |  |      |            |
|------------------------------------|--|------|------------|
| Class description                  | 선택된 날짜에 대한 이벤트를 시간, 내용, 위치로 표시하는 테이블을 생성하는 클래스 |      |            |
| 구분                                 | Name   | Type | Visibility |
|                                    | Description                                    |      |            |
| Attributes                         | columns  | none | private    |
|                                    | 'DataTable' 위젯에서 사용할 각 열에 대한 정의 목록             |      |            |
|                                    | rows   | none | private    |
| 'DataTable' 위젯에서 사용할 각 행에 대한 정의 목록 |  |      |            |

| DataRow           |  |      |            |
|-------------------|--|------|------------|
| Class description | 표의 한 행을 나타내며, 여기에는 해당 행의 각 'DataCell'에 표시될 데이터가 포함되는 클래스 |      |            |
| 구분                | Name   | Type | Visibility |
|                   | Description  |      |            |
| Attributes        | cells  | none | private    |
|                   | 각 행의 데이터 셀을 나타내는 속성                                      |      |            |

| DataCell          |                                   |      |            |
|-------------------|-----------------------------------|------|------------|
| Class description | 'DataTable'에서 표의 한 셀을 나타내는 위젯 클래스 |      |            |
| 구분                | Name                              | Type | Visibility |
|                   | Description                       |      |            |
| Attributes        | Child                             | none | private    |
|                   | 해당 위젯의 단일 자식 위젯을 정의               |      |            |

| Scaffold                           |  |          |            |
|------------------------------------|--|----------|------------|
| Class description                  | 기본적인 구조와 레이아웃을 제공 클래스                                |          |            |
| 구분                                 | Name   | Type     | Visibility |
|                                    | Description  |          |            |
| Attributes                         | appBar   | none     | private    |
|                                    | 앱 상단의 위치하며 제목, 액션 버튼, 메뉴 등을 포함하는 부분                  |          |            |
|                                    | body   | none     | private    |
|                                    | 앱의 주요 내용이 표시되는 영역                                    |          |            |
|                                    | bottomNavigationBar                                  | none     | private    |
| 앱 하단의 위치하며 여러 페이지나 섹션으로 이동하는 탐색 지원 |  |          |            |
| Category                           |  |          |            |
| Class description                  | 'StatefulWidget'을 상속하고 카테고리 페이지를 나타내는 클래스            |          |            |
| 구분                                 | Name   | Type     | Visibility |
|                                    | Description  |          |            |
| Attributes                         | categoryList   | none     | private    |
|                                    | 비동기적으로 데이터를 가져오기 위한 Future 객체를 나타냄                   |          |            |
| 구분                                 | Name   | Argument | Returns    |
|                                    | Description  |          |            |
| Operations                         | initState  | none     | none       |
|                                    | StatefulWidget의 초기 상태를 설정                            |          |            |
|                                    | getPost  | none     | none       |
|                                    | 서버에서 데이터를 가져오는 비동기 메소드                               |          |            |
|                                    | build  | none     | none       |
| 화면을 생성하는 역할                        |  |          |            |
| Future                             |  |          |            |
| Class description                  | 비동기 작업의 결과를 나타내는 클래스                                 |          |            |
| FutureBuilder                      |  |          |            |
| Class description                  | 비동기 작업의 결과를 처리하고 UI를 업데이트하는 위젯 클래스                   |          |            |
| 구분                                 | Name   | Type     | Visibility |
|                                    | Description  |          |            |
| Attributes                         | future   | none     | private    |
|                                    | 비동기 작업을 나타내는 속성                                      |          |            |
|                                    | builder  | none     | private    |
| 결과에 기반하여 UI를 생성하는 콜백 함수            |  |          |            |
| SingleChildScrollView              |  |          |            |
| Class description                  | 하나의 자식 위젯을 가지며, 그 자식이 화면을 벗어날 경우 스크롤이 가능하게 하는 위젯 클래스 |          |            |
| Collmn                             |  |          |            |
| Class description                  | 세로 방향으로 자식 위젯들을 배열하는 레이아웃 위젯 클래스                     |          |            |
| 구분                                 | Name   | Type     | Visibility |
|                                    | Description  |          |            |
| Attributes                         | Container  | none     | private    |
|                                    | 세로로 스크롤 가능한 리스트를 표시                                  |          |            |

| Viewtext                 |   |                                |            |
|--------------------------|---|--------------------------------|------------|
| Class description        | StatelessWidget을 상속받은 위젯으로, 글의 상세 내용을 표시하는 화면을 구현 클래스 |                                |            |
| 구분                       | Name  | Type                           | Visibility |
|                          | Description   |                                |            |
| Attributes               | index   | none                           | private    |
|                          | 해당 화면에서 표시할 글의 인덱스를 받아옴                               |                                |            |
| Attributes               | dataList  | none                           | private    |
|                          | 해당 화면에서 표시할 데이터 리스트를 받아옴                              |                                |            |
| Around                   |   |                                |            |
| Class description        | 'StatefulWidget'을 상속받은 위젯 클래스                         |                                |            |
| 구분                       | Name  | Type                           | Visibility |
|                          | Description   |                                |            |
| Attributes               | locationList  | Future                         | private    |
|                          | 'getPost' 메소드를 호출하여 출력된 결과를 저장                        |                                |            |
| 구분                       | Name  | Argument                       | Returns    |
|                          | Description   |                                |            |
| Operations               | initState   | none                           | none       |
|                          | 위젯이 생성될때 호출되는 초기화 메소드                                 |                                |            |
|                          | getPost   | none                           | none       |
|                          | 주변에 있는 글을 가져오는 비동기 작업을 수행하는 메소드                       |                                |            |
|                          | build   | BuildContext                   | none       |
| 위젯을 화면에 렌더링하는 역할을 하는 메소드 |   |                                |            |
| FutureBuilder            |   |                                |            |
| Class description        | 비동기 작업의 결과에 따라 UI를 동적으로 구성 클래스                        |                                |            |
| 구분                       | Name  | Type                           | Visibility |
|                          | Description   |                                |            |
| Attributes               | future  | locationList                   | private    |
|                          | 객체의 결과를 나타내는 속성                                       |                                |            |
| 구분                       | Name  | Argument                       | Returns    |
|                          | Description   |                                |            |
| Operations               | build   | BuildContext,<br>AsyncSnapshot | none       |
|                          | 위젯을 화면에 렌더링하는 역할을 하는 메소드                              |                                |            |
| ListView.builder         |   |                                |            |
| Class description        | 글 목록을 리스트 형태로 표시 클래스                                  |                                |            |
| 구분                       | Name  | Type                           | Visibility |
|                          | Description   |                                |            |
| Attributes               | itemCount   | dataList.length                | private    |
|                          | 리스트에 표시할 아이템의 총 개수를 지정하는 속성                           |                                |            |
| 구분                       | Name  | Argument                       | Returns    |
|                          | Description   |                                |            |
| Operations               | itemBuilder   | BuildContext, int              | none       |
|                          | 각 아이템을 생성하는 함수를 지정하는 속성                               |                                |            |

| Home                               |   |          |            |
|------------------------------------|---|----------|------------|
| Class description                  | 홈 화면을 구성하고 관리 클래스                               |          |            |
| 구분                                 | Name  | Type     | Visibility |
|                                    | Description                                     |          |            |
| Attributes                         | futureDataList                                  | none     | private    |
|                                    | 데이터 멤버 변수                                       |          |            |
| 구분                                 | Name  | Argument | Returns    |
|                                    | Description                                     |          |            |
| Operations                         | initState                                       | none     | none       |
|                                    | 'getPost()'메소드를 호출하여 'futureDataList'를 설정       |          |            |
|                                    | getPost   | none     | none       |
|                                    | 지정된 URL로 요청을 보내고 성공하면 데이터 반환, 그렇지 않으면 오류 메시지 출력 |          |            |
|                                    | build   | none     | none       |
|                                    | 위젯 트리를 정의하고 위젯의 시각적 표현 반환                       |          |            |
| Future                             |   |          |            |
| Class description                  | 비동기 작업의 결과를 나타내는 클래스                            |          |            |
| Scaffold                           |   |          |            |
| Class description                  | 기본적인 구조와 레이아웃을 제공 클래스                           |          |            |
| 구분                                 | Name  | Type     | Visibility |
|                                    | Description                                     |          |            |
| Attributes                         | appBar  | none     | private    |
|                                    | 앱 상단의 위치하며 제목, 액션 버튼, 메뉴 등을 포함하는 부분             |          |            |
|                                    | body  | none     | private    |
|                                    | 앱의 주요 내용이 표시되는 영역                               |          |            |
|                                    | bottomNavigationBar                             | none     | private    |
| bottomNavigationBar                |   |          |            |
| 액 하단의 위치하며 여러 페이지나 섹션으로 이동하는 탐색 지원 |   |          |            |
| FutureBuilder                      |   |          |            |
| Class description                  | 비동기 작업의 결과에 따라 UI를 동적으로 구성 클래스                  |          |            |
| 구분                                 | Name  | Type     | Visibility |
|                                    | Description                                     |          |            |
| Attributes                         | future  | none     | private    |
|                                    | 비동기적으로 실행되는 작업을 정의                              |          |            |
|                                    | builder   | none     | private    |
|                                    | 작업이 완료된 후 UI를 빌드하기 위한 콜백 함수                     |          |            |
| SingleChildScrollView              |   |          |            |
| Class description                  | 자식 위젯이 화면보다 큰 경우 스크롤이 가능한 스크롤 영역 제공 클래스         |          |            |

| ListTile                      |                                     |                |            |
|-------------------------------|-------------------------------------|----------------|------------|
| Class description             | 터치 시 글의 상세 내용을 보여주는 화면으로 이동 클래스     |                |            |
| 구분                            | Name                                | Type           | Visibility |
|                               | Description                         |                |            |
| Attributes                    | contentPadding                      | Edgeinsets     | private    |
|                               | ' ListTile '의 내용에 대한 안쪽 여백을 설정하는 속성 |                |            |
|                               | title                               | Align          | private    |
|                               | 리스트 아이템의 주요 제목                      |                |            |
|                               | subtitle                            | Column         | private    |
|                               | 리스 아이템의 부가적인 정보                     |                |            |
|                               | onTap                               | Navigator.push | private    |
| 리스트 아이템을 터치했을 때 실행되는 콜백 함수 지정 |                                     |                |            |

| Write             |                    |  |  |
|-------------------|--------------------|--|--|
| Class description | 글을 작성하는 페이지 버튼 클래스 |  |  |

| SizedBox          |                |      |            |
|-------------------|----------------|------|------------|
| Class description | 위젯의 크기를 조절 클래스 |      |            |
| 구분                | Name           | Type | Visibility |
|                   | Description    |      |            |
| Attributes        | child          | none | private    |
|                   | 위젯의 단일 자식을 나타냄 |      |            |

| Info              |                           |          |         |
|-------------------|---------------------------|----------|---------|
| Class description | 정보 페이지의 UI 및 동작을 정의하는 클래스 |          |         |
| 구분                | Name                      | Argument | Returns |
|                   | Description               |          |         |
| Operations        | _member_Withdrawal        | none     | none    |
|                   | 회원 탈퇴를 수행하는 메소드           |          |         |
|                   | _withdrawal_dialog        | none     | none    |
|                   | 회원 탈퇴를 확인하는ダイ얼로그를 띠우는 메소드 |          |         |
|                   | _check_dialog             | none     | none    |
|                   | 로그아웃을 확인하는ダイ얼로그를 띠우는 메소드  |          |         |
|                   | build                     | none     | none    |
| 위젯을 화면에 랜더링하는 역할  |                           |          |         |

| Scaffold          |                     |      |            |
|-------------------|---------------------|------|------------|
| Class description | 앱의 기본 구조를 제공하는 클래스  |      |            |
| 구분                | Name                | Type | Visibility |
|                   | Description         |      |            |
| Attributes        | appBar              | none | private    |
|                   | 화면 상단에 앱 바를 표시하는 속성 |      |            |
|                   | body                | none | private    |
|                   | 화면의 주요 내용을 포함하는 속성  |      |            |

| User_mo           |                  |          |         |
|-------------------|------------------|----------|---------|
| Class description | 사용자 모드 정의 클래스    |          |         |
| 구분                | Name             | Argument | Returns |
|                   | Description      |          |         |
| Operations        | build            | none     | none    |
|                   | 위젯을 화면에 랜더링하는 역할 |          |         |

| MaterialApp       |   |      |            |
|-------------------|---|------|------------|
| Class description | Material Design을 따르는 앱의 기본 구조를 설정하는 클래스 |      |            |
| 구분                | Name                                    | Type | Visibility |
|                   | Description                             |      |            |
| Attributes        | home                                    | none | private    |
|                   | 첫 번째 화면을 정의                             |      |            |

| Login             |                  |          |         |
|-------------------|------------------|----------|---------|
| Class description | 로그인 화면을 정의하는 클래스 |          |         |
| 구분                | Name             | Argument | Returns |
|                   | Description      |          |         |
| Operations        | build            | none     | none    |
|                   | 위젯을 화면에 랜더링하는 역할 |          |         |

### 3.2.4 homestate

| Hobbys            |             |              |            |
|-------------------|-------------|--------------|------------|
| Class description | 상위 취미 클래스   |              |            |
| 구분                | Name        | Type         | Visibility |
|                   | Description |              |            |
| Attributes        | Hobbys      | List<String> | private    |
|                   | 상위 취미 리스트   |              |            |

| MyHobbyList       |               |              |            |
|-------------------|---------------|--------------|------------|
| Class description | 내 취미 저장하는 클래스 |              |            |
| 구분                | Name          | Type         | Visibility |
|                   | Description   |              |            |
| Attributes        | myHobby       | List<String> | private    |
|                   | 현재 저장된 취미 리스트 |              |            |

| DetailHobbys      |              |                    |            |
|-------------------|--------------|--------------------|------------|
| Class description | 하위 취미 클래스    |                    |            |
| 구분                | Name         | Type               | Visibility |
|                   | Description  |                    |            |
| Attributes        | detailHobbys | List<List<String>> | private    |
|                   | 세부 취미 리스트    |                    |            |

| HobbyApp          |                 |          |         |
|-------------------|-----------------|----------|---------|
| Class description | 내 정보에 취미 저장 클래스 |          |         |
| 구분                | Name            | Argument | Returns |
|                   | Description     |          |         |
| Operations        | runApp          | none     | void    |
|                   | 취미 저장을 실행하는 메소드 |          |         |

| <<class>> _Login                 |                                    |                       |            |
|----------------------------------|------------------------------------|-----------------------|------------|
| Class description                | 사용자가 앱에 로그인하는 화면을 구현하는 클래스         |                       |            |
| 구분                               | Name                               | Type                  | Visibility |
|                                  | Description                        |                       |            |
| Attributes                       | ID                                 | TextEditingController | private    |
|                                  | 사용자 아이디                            |                       |            |
| Operations                       | Password                           | TextEditingController | private    |
|                                  | 사용자 비밀번호                           |                       |            |
| 구분                               | Name                               | Argument              | Returns    |
|                                  | Description                        |                       |            |
| Operations                       | login                              | none                  | void       |
|                                  | 사용자가 입력한 아이디와 비밀번호를 서버로 전송 및 결과 동작 |                       |            |
|                                  | _showDialog                        | BuildContext          | void       |
|                                  | 아이디 또는 비밀번호가 잘못되었을 때 경고 표시하는 메소드   |                       |            |
|                                  | build                              | BuildContext          | Widget     |
| 사용자에게 아이디와 비밀번호를 입력받을 수 있는 폼을 제공 |                                    |                       |            |

| <<class>> _RegistrationState |                                  |                       |            |
|------------------------------|----------------------------------|-----------------------|------------|
| Class description            | 'Registration' 클래스의 상태를 나타내는 클래스 |                       |            |
| 구분                           | Name                             | Type                  | Visibility |
|                              |                                  | Description           |            |
| Attributes                   | Email                            | TextEditingController | private    |
|                              |                                  | 사용자 이메일               |            |
| Operations                   | Password                         | TextEditingController | package    |
|                              |                                  | 사용자 비밀번호              |            |
| 구분                           | Name                             | Argument              | Returns    |
|                              |                                  | Description           |            |
| Operations                   | register                         | none                  | void       |
|                              |                                  | 회원가입                  |            |
|                              | build                            | BuildContext          | Widget     |
|                              |                                  | 위젯의 화면을 렌더링하는 메소드     |            |

| <<class>> _HomeState |                               |  |  |
|----------------------|-------------------------------|--|--|
| Class description    | 앱의 홈 화면을 관리하는 StateWidget 클래스 |  |  |

| <<class>> _Pick   |                               |               |            |
|-------------------|-------------------------------|---------------|------------|
| Class description | 사용자가 취미를 선택하는 화면을 구현하는 클래스    |               |            |
| 구분                | Name                          | Type          | Visibility |
|                   |                               | Description   |            |
| Attributes        | _tabController                | TabController | private    |
|                   | 리스트의 길이를 이용하여 취미 카테고리제어하는 속성  |               |            |
| Operations        | current_index                 | int           | private    |
|                   | 현재 선택된 취미 카테고리의 인덱스 저장하는 변수   |               |            |
| Attributes        | pick_hobby                    | String        | private    |
|                   | 사용자가 선택한 취미를 저장하는 문자열 변수      |               |            |
| Operations        | index                         | int           | private    |
|                   | 현재까지 선택한 취미의 수를 저장하는 변수       |               |            |
| Attributes        | max_select                    | int           | private    |
|                   | 사용자가 선택할 수 있는 취미 수를 나타내는 변수   |               |            |
| 구분                | Name                          | Argument      | Returns    |
|                   |                               | Description   |            |
| Operations        | initState                     | none          | void       |
|                   | 메소드는 State 객체가 생성될 때 호출되는 메소드 |               |            |
| Attributes        | dispose                       | none          | void       |
|                   | State 객체가 파기될 때 호출되는 메소드      |               |            |
| Operations        | add_hobby                     | String        | void       |
|                   | 취미를 추가하거나 제거하는 메소드            |               |            |
| Operations        | build                         | BuildContext  | Widget     |
|                   | 화면을 그리는 역할을 하는 메소드            |               |            |

| <<class>> _Registration_ |                                    |                       |            |
|--------------------------|------------------------------------|-----------------------|------------|
| Class description        | 회원 가입을 담당하는 화면의 상태를 관리하는 클래스       |                       |            |
| 구분                       | Name                               | Type                  | Visibility |
|                          | Description                        |                       |            |
| Attributes               | ID_Controller                      | TextEditingController | private    |
|                          | 아이디를 입력받기 위한 텍스트 컨트롤러              |                       |            |
|                          | Password_Controller                | TextEditingController | private    |
|                          | 비밀번호를 입력받기 위한 텍스트 컨트롤러             |                       |            |
|                          | Name_Controller                    | TextEditingController | private    |
|                          | 이름을 입력받기 위한 텍스트 컨트롤러               |                       |            |
|                          | Nickname_Controller                | TextEditingController | private    |
|                          | 닉네임을 입력받기 위한 텍스트 컨트롤러              |                       |            |
|                          | id_check                           | bool                  | private    |
|                          | 아이디 중복 체크하는 변수                     |                       |            |
|                          | nic_check                          | bool                  | private    |
|                          | 닉네임 중복 체크하는 변수                     |                       |            |
| Operations               | is_company                         | bool                  | private    |
|                          | 가입 회원 여부를 나타내는 체크박스의 상태 저장하는 변수    |                       |            |
|                          | 구분                                 | Name                  | Argument   |
|                          | Description                        |                       |            |
|                          | _showDialog                        | BuildContext          | void       |
|                          | 입력 필드가 비어있을 때 경고창을 띄우는 메소드         |                       |            |
|                          | _check_available_id_Dialog         | BuildContext          | void       |
|                          | 아이디 또는 닉네임 중복 체크를 하지 않았을 때 나오는 경고창 |                       |            |
|                          | _available_id_Dialog               | BuildContext          | void       |
|                          | 아이디가 사용 가능할 때 나오는 알림창              |                       |            |
|                          | _available_nic_Dialog              | BuildContext          | void       |
|                          | 닉네임이 사용 가능할 때 나오는 알림창              |                       |            |
|                          | _refuse_id_Dialog                  | BuildContext          | void       |
|                          | 아이디가 사용 불가능할 때 나오는 경고창             |                       |            |
|                          | _refuse_nic_Dialog                 | BuildContext          | void       |
|                          | 닉네임이 사용 불가능할 때 나오는 경고창             |                       |            |
|                          | _checkd_id_uplication              | none                  | void       |
|                          | 아이디의 중복 여부를 체크하는 메소드               |                       |            |
|                          | _checkd_nic_uplication             | none                  | void       |
|                          | 닉네임의 중복 여부를 체크하는 메소드               |                       |            |
|                          | build                              | BuildContext          | Widget     |
|                          | UI를 구성하고 반환하는 메소드                  |                       |            |

| <<class>> _Hobby_ |   |               |            |
|-------------------|---|---------------|------------|
| Class description | Hobby 위젯의 상태를 관리하는 클래스                    |               |            |
| 구분                | Name                                      | Type          | Visibility |
|                   | Description                               |               |            |
| Attributes        | _tabController                            | TabController | private    |
|                   | 'TabController' 객체로 Hobbys의 길이를 탭의 개수로 설정 |               |            |
|                   | current_index                             | int           | private    |
|                   | 현재 선택된 상위 취미의 인덱스                         |               |            |
|                   | pick_hobby                                | String        | private    |
|                   | 선택된 취미를 저장하는 문자열                          |               |            |
|                   | index                                     | int           | private    |
|                   | 현재 선택된 취미의 개수                             |               |            |
|                   | max_select                                | int           | private    |
|                   | 취미의 최대 개수를 제한하기 위한 변수                     |               |            |
| Operations        | selectedHobbies                           | List<bool>    | private    |
|                   | 각 취미가 선택되었는지 여부를 나타내는 변수                  |               |            |
|                   | Name                                      | Argument      | Returns    |
|                   | Description                               |               |            |
|                   | initState                                 | none          | void       |
|                   | 'TabController' 객체를 초기화하는 메소드             |               |            |
|                   | dispose                                   | none          | void       |
|                   | 'TabController' 객체를 해제하는 메소드              |               |            |
|                   | add_hobby                                 | String        | void       |
|                   | 취미를 추가하거나 삭제하는 메소드                        |               |            |

| <<class>> _Local_ |                           |                |            |
|-------------------|---------------------------|----------------|------------|
| Class description | Local 위젯의 상태를 관리하는 클래스    |                |            |
| 구분                | Name                      | Type           | Visibility |
|                   | Description               |                |            |
| Attributes        | current_index             | int            | private    |
|                   | 현재 선택된 상위 지역의 인덱스         |                |            |
|                   | pick_local                | String         | private    |
|                   | 선택된 지역을 저장하는 문자열          |                |            |
|                   | index                     | int            | private    |
|                   | 현재 선택된 지역의 개수             |                |            |
|                   | max_select                | int            | private    |
|                   | 선택 가능한 지역의 최대 개수한하기 위한 변수 |                |            |
|                   | iindex                    | String         | private    |
|                   | 선택된 지역을 식별하기 위한 문자열       |                |            |
| Operations        | Name                      | Argument       | Returns    |
|                   | Description               |                |            |
|                   | add_local                 | String, String | void       |
|                   | 지역을 추가하거나 삭제하는 메소드        |                |            |
|                   | register                  | none           | void       |
|                   | 사용자 정보를 서버에 등록하는 메소드      |                |            |
|                   | build                     | BuildContext   | Widget     |
|                   | UI를 구성하고 반환하는 메소드         |                |            |

## 4. Sequence diagram

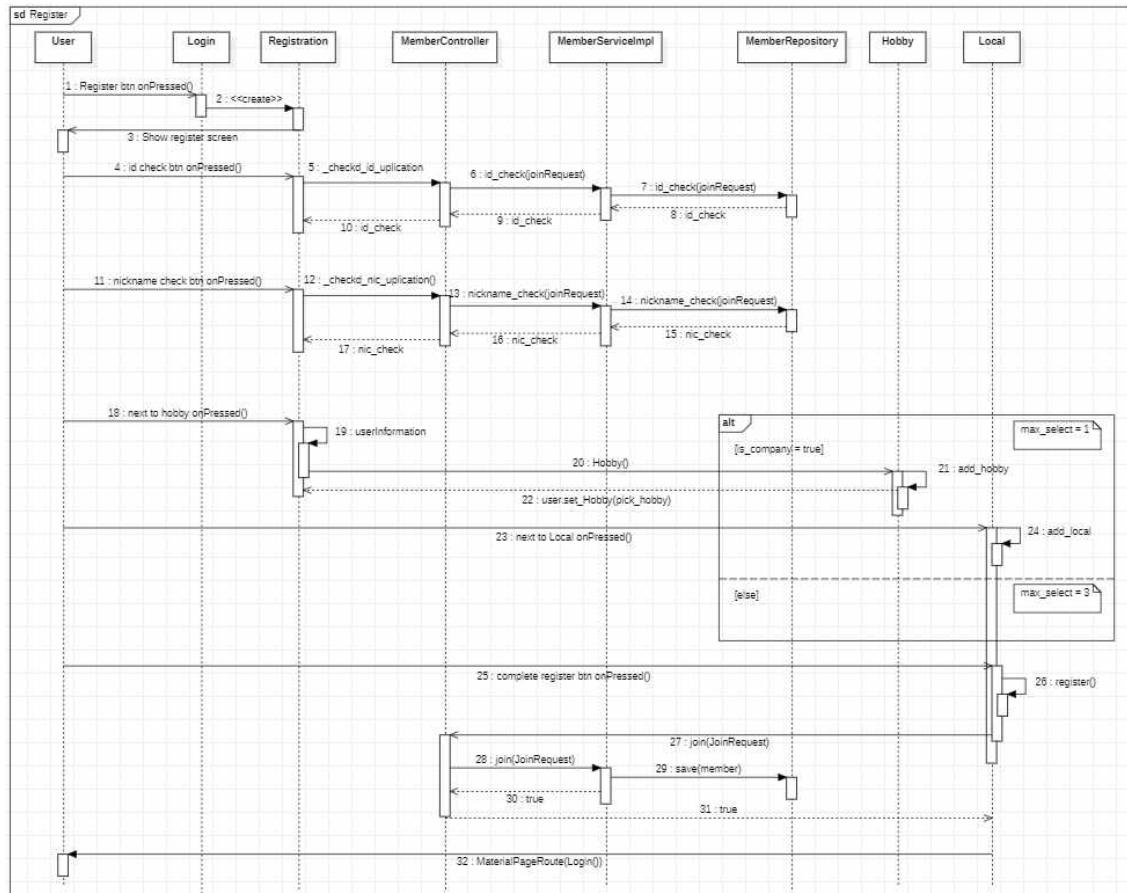
이 섹션에서는 Sequence Diagrams(이하 SD)와 각 SD에 대한 자세한 설명을 제공한다. 앞서 서술한 use case와 class diagram을 토대로 만들었으며, StarUML을 활용하여 작성하였다.

SD에 관한 고려 사항은 다음과 같다.

1. Accept Invitation SD를 제외한 모든 SD는 특정 use case와 대응되며, main/extension success scenario의 흐름을 따른다.
2. SD 맨 왼쪽에 있는 User Lifeline은 실제 코드 속 class가 아닌, 시스템 사용자를 의미한다.
3. Lifeline 중 하나인 MemberRepository, CommentRepository는 클래스가 아닌 인터페이스이다. 그러나 DB와 ServiceImpl 클래스 간의 상호 작용을 담당하는 중요한 인터페이스이기 때문에 Lifeline으로 작성하였다.
4. User와 Repository, AppBar를 제외한 모든 SD의 생명선은 class 이름을 적은 것이며, 객체는 따로 표기하지 않았다.
5. HTTP request가 있는 메소드들이 있다. 이는 메시지로 메소드 이름을 그대로 적어주되, (send post request)라고 따로 표기했다.
6. Register SD의 alt 프레임은 개인 회원과 기업 회원을 구분 짓는 프레임이다. 기업 회원은 취미와 위치를 최대 1개까지만 선택할 수 있기 때문에 max\_select = 1라고 표기하고, 개인 회원은 취미와 위치를 최대 3개까지만 선택할 수 있기 때문에 max\_select = 3이라고 표기했다.
7. Withdrawal(Admin)의 SD는 관리자가 관리자 계정에 로그인한 상태라는 가정하에 SD를 표현한다.
8. View comment SD는 View post SD와 흐름이 동일하기 때문에 생략한다.
9. 다른 사용자와 상호작용하는 SD일 경우, 다른 사용자의 글과 댓글이 최소한 하나 이상 있다는 가정하에 설명한다.
10. Use case #16 Manage My profile에선 기능 두 가지(프로필 보기, 수정하기)를 합쳐서 서술하였으나, SD에서는 두 기능을 나누어 설명한다.
11. Accept Invitation SD는 Use case #13-5에 해당하는 기능으로, Use case는 아니지만 일정 추가라는 기능을 설명하기 위해 따로 설명한다.

## 4.1 User sequence diagram

### 4.1.1 Register



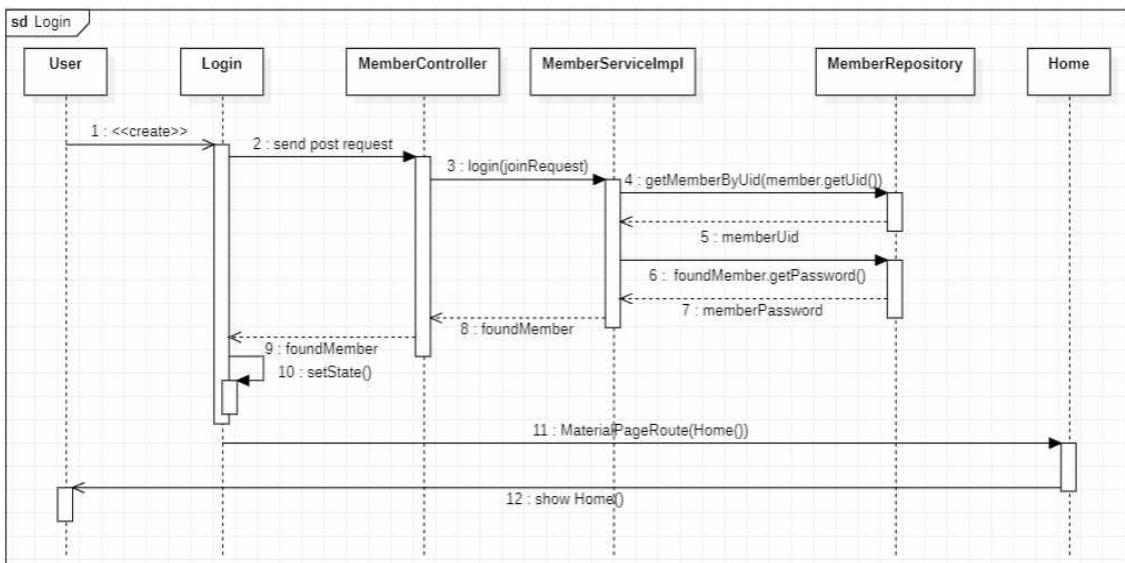
[그림 4-1] Register SD

사용자가 시스템에 회원가입 하는 use case를 나타내는 sequence diagram이다. <Use case #1>에 해당하며, 회원가입의 순서는 다음과 같다.

- ① 사용자가 회원가입 버튼을 누르면 회원가입 화면이 출력된다.
- ② 회원가입 화면에는 사용자의 정보(아이디, 비밀번호, 이름, 닉네임)를 입력받는 창과 회원의 종류를 선택할 수 있는 체크박스가 표시된다.
- ③ 사용자는 사용자의 정보를 모두 입력한다. 이때, 아이디와 닉네임은 중복 확인 절차가 필요하다.
- ④ 사용자는 회원의 종류를 선택한다. 체크박스를 누를 경우 기업 회원으로, 누르지 않을 경우 일반 회원으로 가입할 수 있다.
- ⑤ 사용자는 회원가입 화면에서 다음 버튼을 누른다.
- ⑥ 사용자가 입력했던 정보가 DB에 저장된 후, 취미 선택 화면이 출력된다. 일반 회원은 최대 3개, 기업 회원은 최대 1개의 취미를 선택할 수 있다.

- ⑥ 사용자는 취미 선택 화면에서 다음 버튼을 누른다.
- ⑦ 사용자가 선택한 취미가 DB에 저장된 후, 지역 설정 화면이 출력된다. 일반 회원은 최대 3개, 기업 회원은 최대 1개의 지역을 선택할 수 있다.
- ⑦ 지역 설정 후 완료 버튼을 누르면 서버에 회원가입 요청을 하고, 회원가입에 성공할 경우 로그인 화면으로 전환된다.

#### 4.1.2 Login

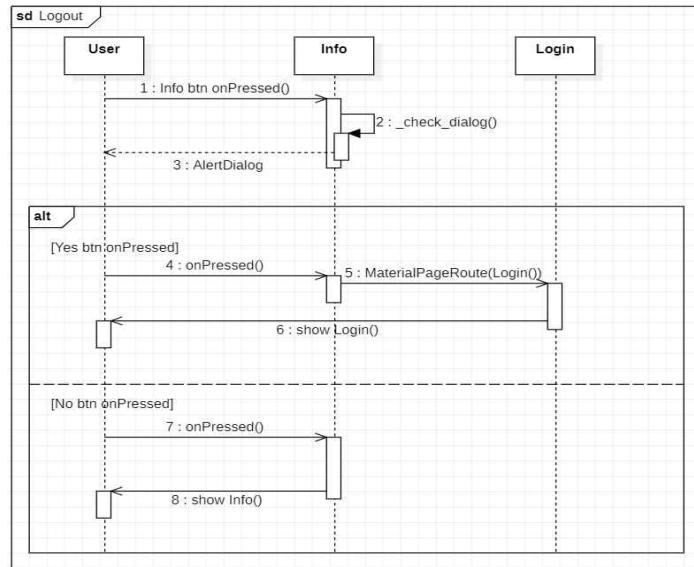


[그림 4-2] Login SD

사용자가 시스템에 로그인하는 use case를 나타내는 sequence diagram이다. <Use case #2>에 해당하며, 로그인의 순서는 다음과 같다.

- ① 사용자가 자신의 아이디와 비밀번호를 입력한 후 로그인 버튼을 누른다.
- ② 시스템은 로그인 요청을 받아 사용자가 입력한 정보가 DB에 있는지 확인한다.
- ③ 계정이 있다면 DB에 저장된 계정 정보를 가져와 로그인에 성공시킨다. 계정이 없다면 로그인 버튼을 눌러도 홈 화면으로 전환되지 않는다.

#### 4.1.3 Logout

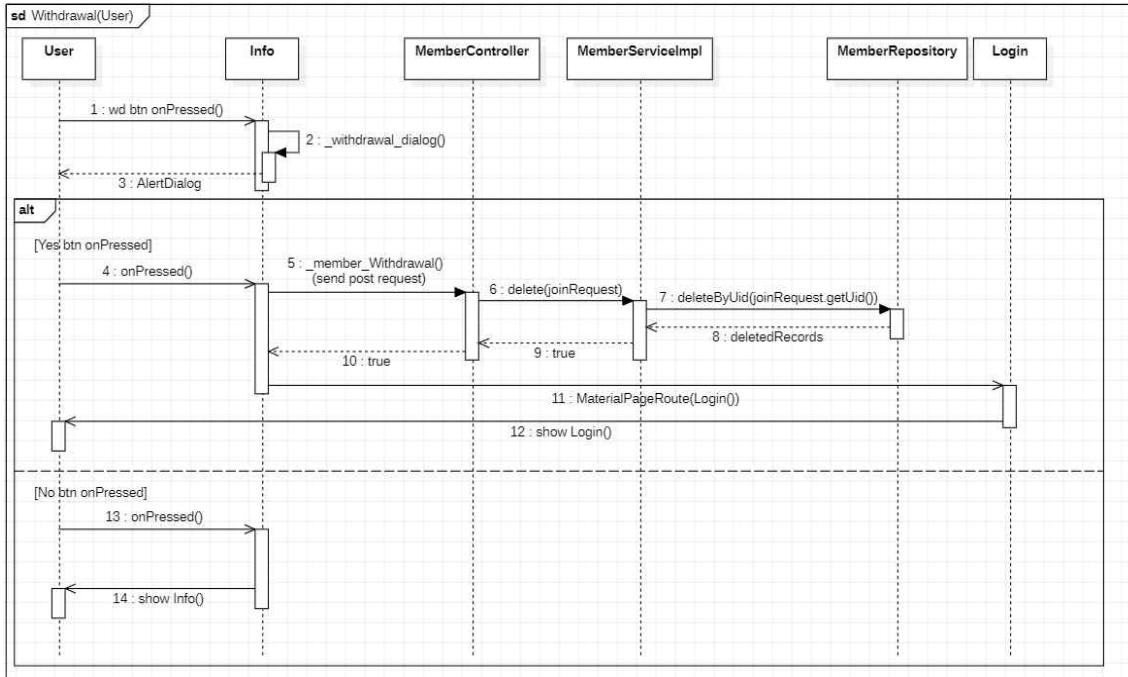


[그림 4-3] Logout SD

사용자가 시스템에 로그아웃하는 use case를 나타내는 sequence diagram이다. <Use case #3>에 해당하며, 로그아웃의 순서는 다음과 같다.

- ① 사용자가 내 정보 탭을 누르면 내 프로필 화면이 출력된다.
- ② 사용자가 내 프로필 화면에 표시된 로그아웃 버튼을 누르면, 로그아웃 여부를 묻는 로그아웃 확인창이 출력된다.
- ③ 사용자는 로그아웃 확인창에서 네, 아니오 중 하나를 선택할 수 있다.
- ④ 사용자가 네 버튼을 누르면 로그아웃 기능이 시작된다. 로그아웃에 성공했을 경우 내 프로필 화면이 로그인 화면으로 전환된다.
- ⑤ 사용자가 아니오 버튼을 누르면 로그아웃 확인창이 꺼지고 내 프로필 화면으로 돌아간다.

#### 4.1.4 Withdrawal(User)

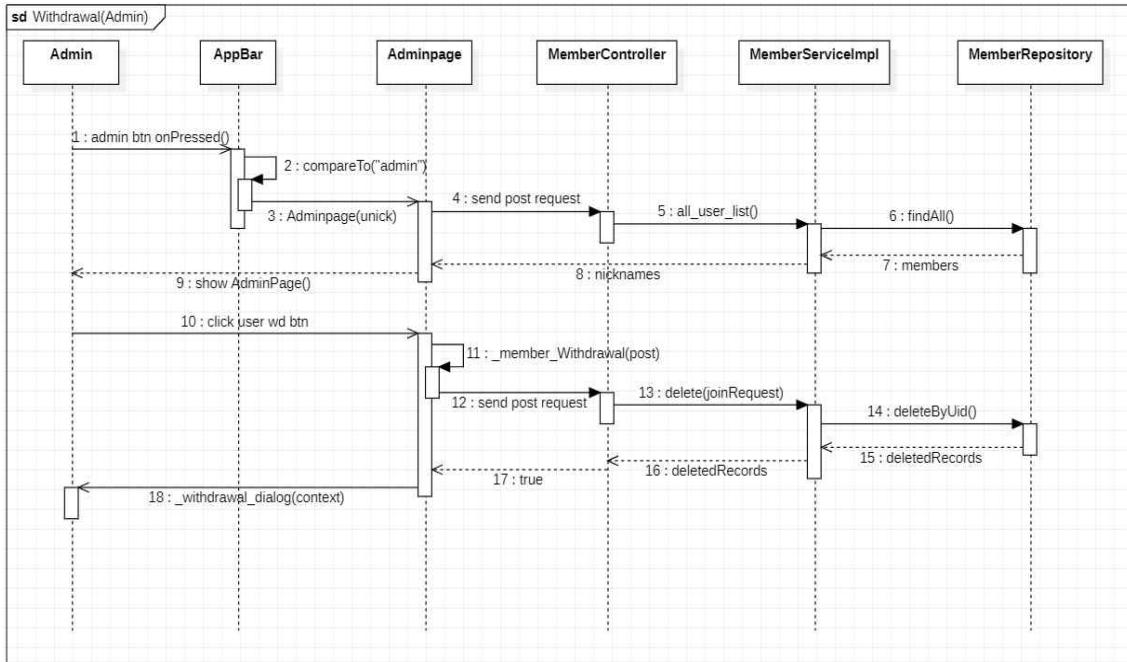


[그림 4-4] Withdrawal(User) SD

사용자가 시스템을 탈퇴하는 use case를 나타내는 sequence diagram이다. <Use case #17>에 해당하며, 시스템 탈퇴의 순서는 다음과 같다.

- ① 사용자가 내 프로필 화면에 표시된 회원 탈퇴 버튼을 누르면, 탈퇴 여부를 묻는 탈퇴 확인창이 출력된다.
- ② 사용자는 탈퇴 확인창에서 네, 아니오 중 하나를 선택할 수 있다.
- ③ 사용자가 네 버튼을 누르면 서버에 회원 탈퇴 요청을 보낸다. 탈퇴에 성공했을 경우 deletedRecords를 반환하여 사용자의 계정은 DB에서 삭제되고, 내 프로필 화면이 로그인 화면으로 전환된다.
- ④ 사용자가 아니오 버튼을 누르면 탈퇴 확인창이 꺼지고 내 프로필 화면으로 돌아간다.

#### 4.1.4 Withdrawal(Admin)



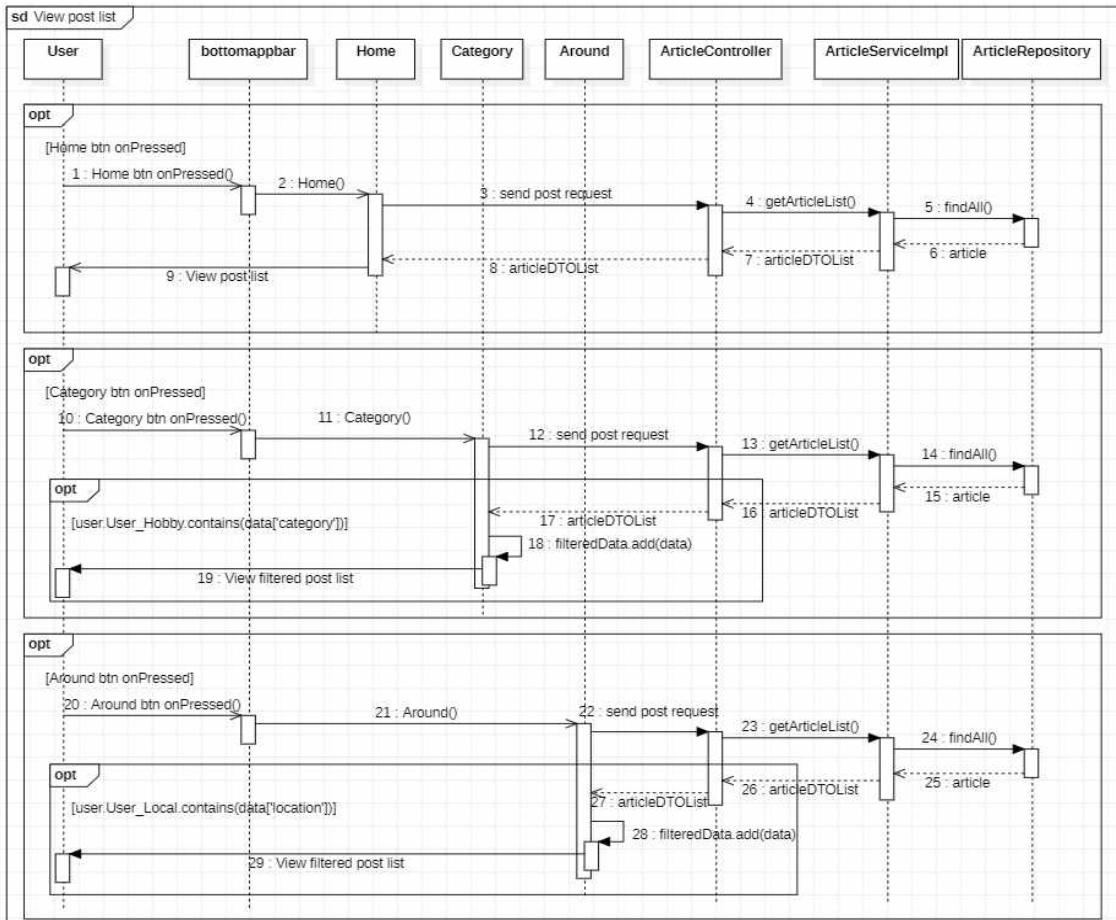
[그림 4-5] Withdrawal(Admin) SD

관리자가 사용자를 탈퇴시키는 use case를 나타내는 sequence diagram이다. <Use case #18>에 해당하며, 순서는 다음과 같다.

- ① 관리자가 상단 바에 있는 사용자 관리 버튼을 누른다.
- ② 시스템에선 버튼을 누른 사용자가 관리자인지 확인한 후 관리자 페이지를 띄운다. 관리자 페이지에 모든 사용자들의 닉네임을 출력한다.
- ③ 관리자는 탈퇴시키고자 하는 사용자의 닉네임 옆 버튼을 누른 뒤 사용자 삭제 버튼을 눌러 관리자를 탈퇴시킨다.
- ④ 시스템은 사용자를 탈퇴시킨 뒤 성공 문구를 출력한다.

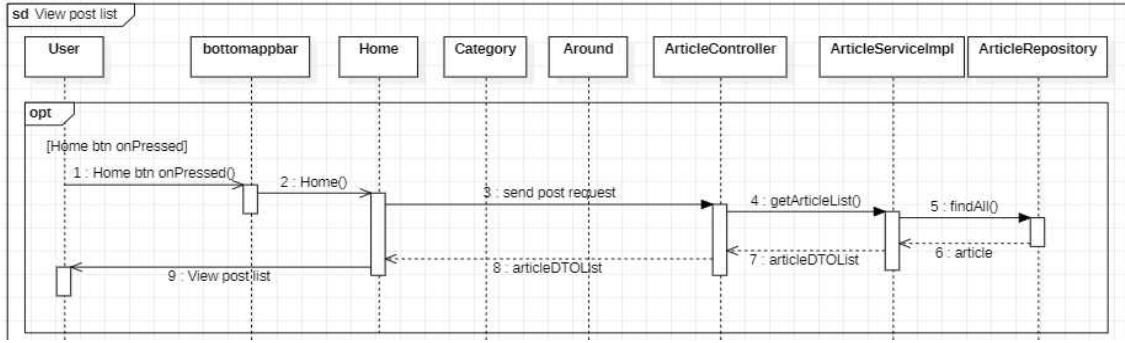
## 4.2 Post sequence diagram

### 4.2.1 View post list



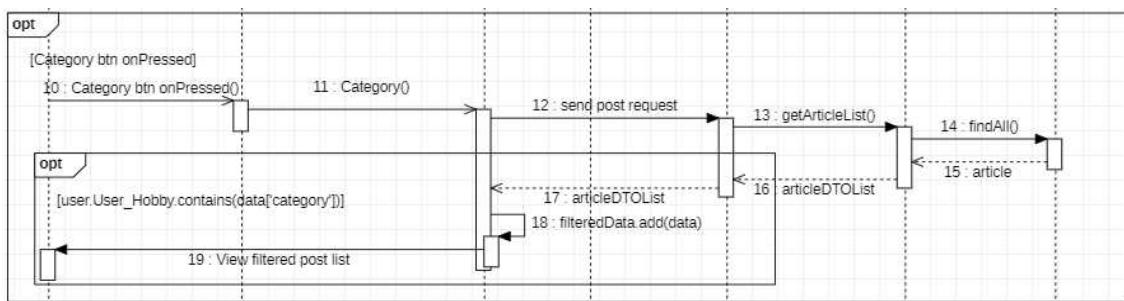
[그림 4-6] View post list SD

사용자가 글 목록을 보는 use case를 나타내는 sequence diagram이다. <Use case #4>에 해당한다. View post list SD는 사용자가 하단 바에 있는 홈, 카테고리, 주변 취미 버튼을 눌렀을 때 시작되는 기능이다. 각 버튼을 눌렀을 때의 SD를 자세히 살펴보자.



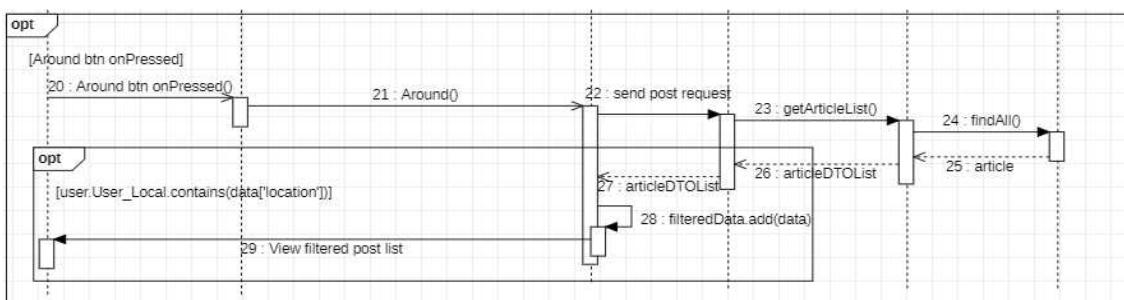
[그림 4-6-1] View Home post list SD

- ① 사용자가 하단 바 속 홈 버튼을 누르면 기능이 시작된다.
- ② 서버에 요청을 보내 모든 글 목록을 불러온 후 출력한다.



[그림 4-6-2] View Category post list SD

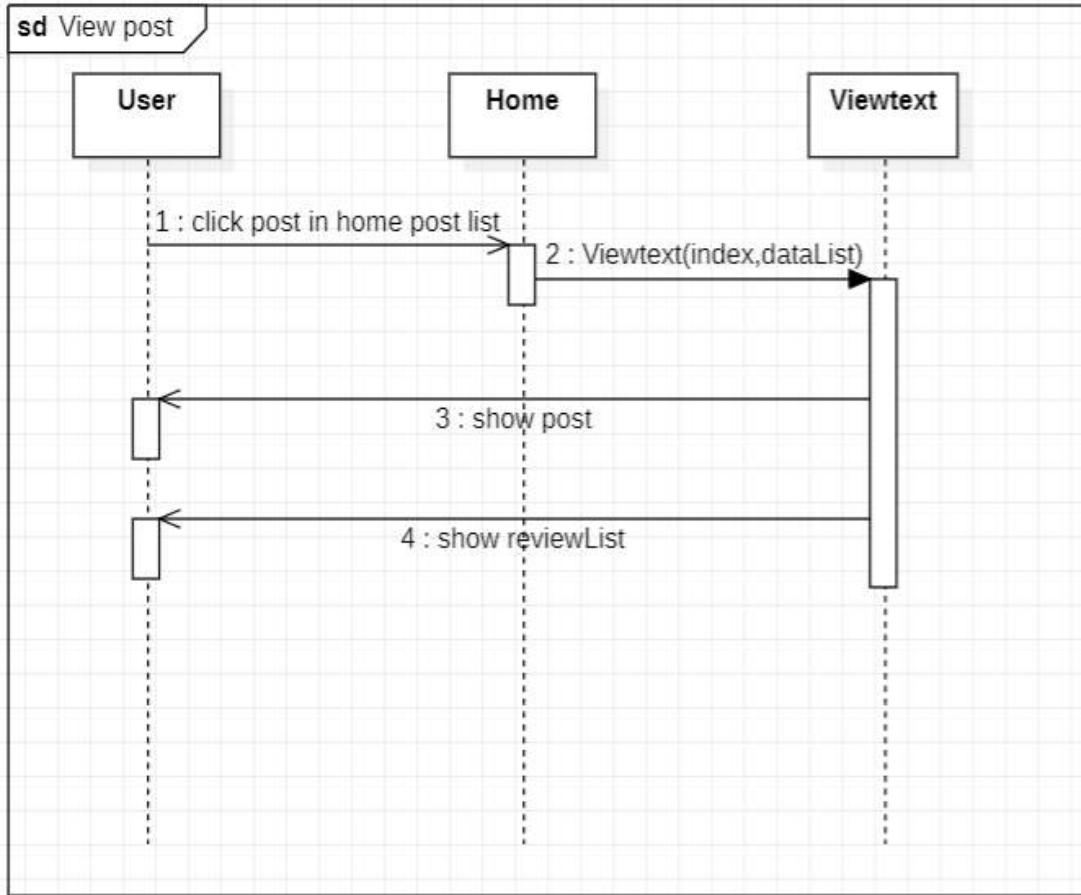
- ① 사용자가 하단 바 속 카테고리 버튼을 누르면 기능이 시작된다.
- ② 서버에 요청을 보내 모든 글 목록을 불러온다.
- ③ 사용자가 설정한 카테고리에 속하는 글을 골라 filteredData에 저장한다.
- ④ 카테고리로 필터링 된 글 목록을 출력한다.



[그림 4-6-3] View Around post list SD

- ① 사용자가 하단 바 속 주변 취미 버튼을 누르면 기능이 시작된다.
- ② 서버에 요청을 보내 모든 글 목록을 불러온다.
- ③ 사용자가 설정한 지역에 속하는 글을 골라 filteredData에 저장한다.
- ④ 지역으로 필터링 된 글 목록을 출력한다.

#### 4.2.2 View post

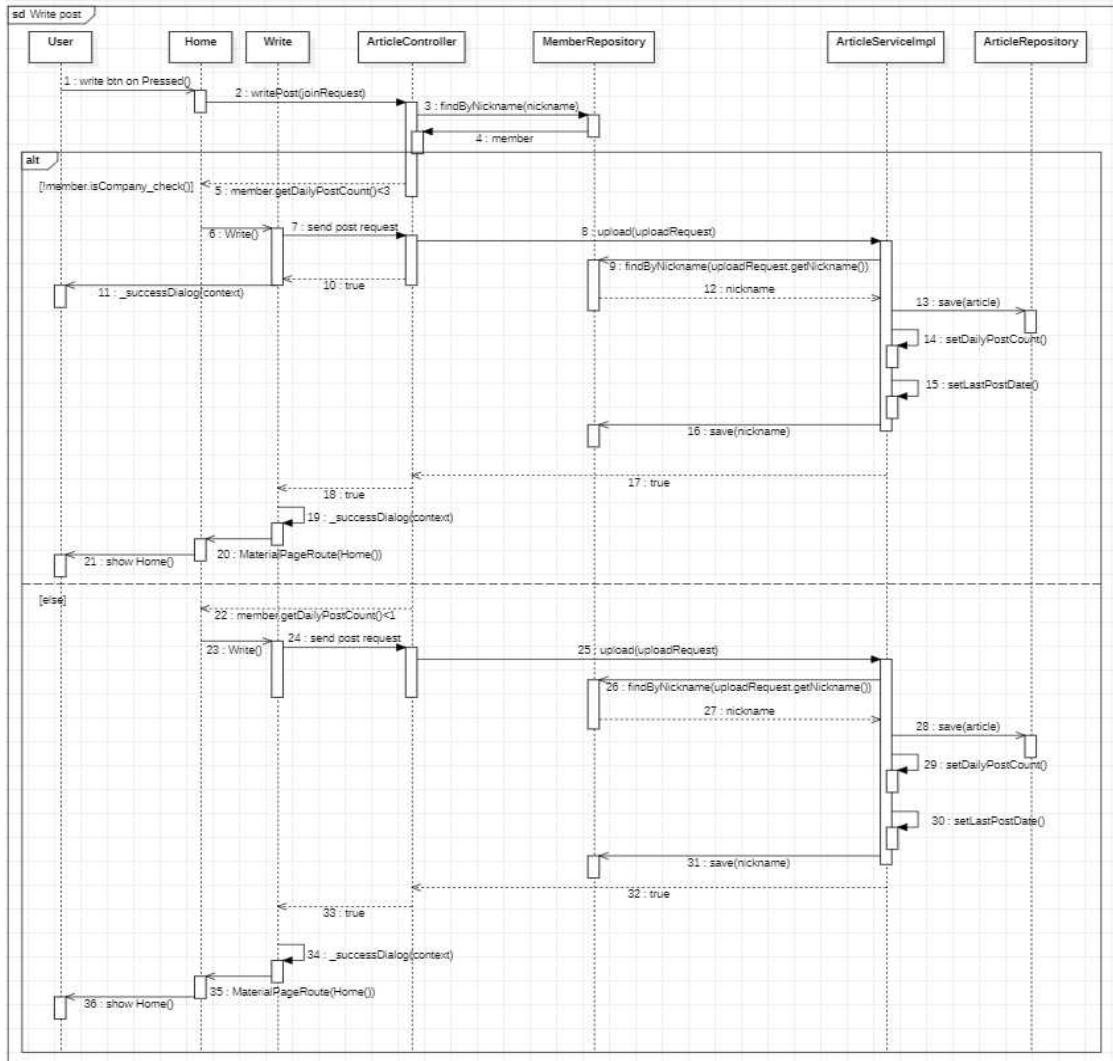


[그림 4-7] View post SD

사용자가 글을 보는 use case를 나타내는 sequence diagram이다. <Use case #5>에 해당한다. 앞서 설명한 View post list SD의 홈, 카테고리, 주변 취미 버튼을 눌러 글 목록을 본 뒤 사용자가 글을 직접 눌렀을 때 시작하는 기능이다. 이 SD에선 홈 버튼을 눌러 글 목록을 출력한 이후의 상황을 서술한다.

- ① 사용자가 글 목록에서 원하는 글을 누른다.
- ② 홈 화면에서 글 보기(Viewtext) 화면으로 넘어가면서 View post 기능이 시작된다.
- ③ 글 보기 기능은 별도의 서버 접근이 필요하지 않으며, 글과 댓글을 같이 출력한다.
- ④ 사용자는 글 제목, 글 내용, 작성자 닉네임, 댓글 목록을 모두 확인할 수 있다.

### 4.2.3 Write post

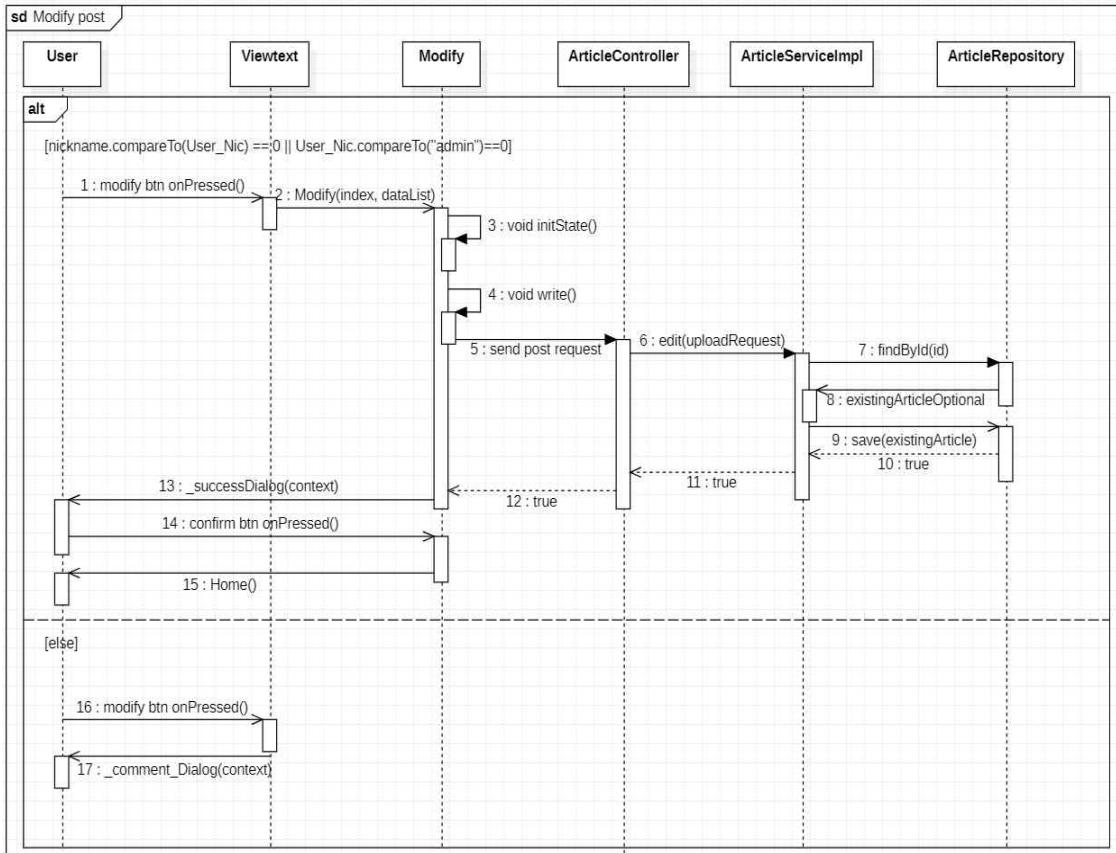


[그림 4-8] Write post SD

사용자가 글을 작성하는 use case를 나타내는 sequence diagram이다. <Use case #6>에 해당한다. 글쓰기 기능은 사용자가 홈 화면에서 글쓰기 버튼을 눌렀을 때 시작된다. 따라서 이 SD는 사용자가 글쓰기 버튼을 누르는 시점부터 서술한다.

- ① 사용자가 홈 화면에서 글쓰기 버튼을 누른다.
- ② 서버에 요청해 사용자의 닉네임을 찾아 어떤 종류의 회원인지 확인한다.
- ③ 기업 회원은 하루에 글 1개, 개인 회원은 하루에 글 3개를 작성할 수 있다. Write post SD에선 글 작성에 성공할 경우만 서술한다.
- ④ 사용자는 글 내용을 모두 입력하고 업로드한다.
- ⑤ 글 업로드에 성공했을 경우, 성공 문구를 화면에 출력한다.

#### 4.2.4 Modify post

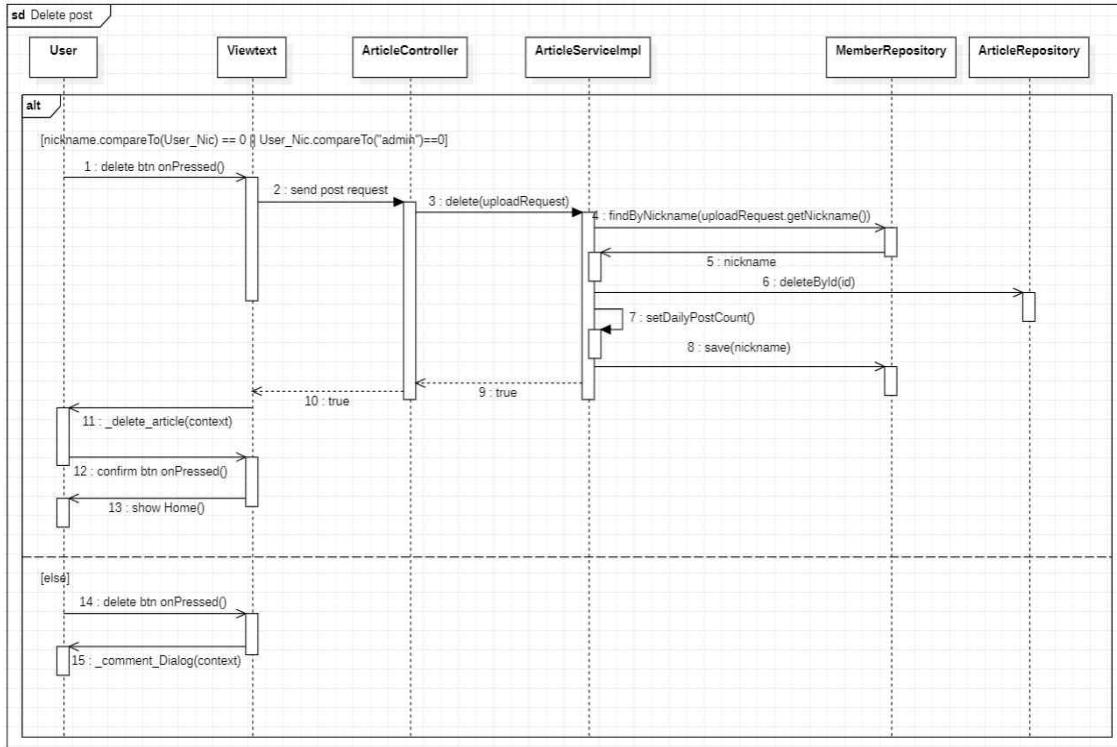


[그림 4-9] Modify post SD

사용자가 글을 수정하는 use case를 나타내는 sequence diagram이다. <Use case #7>에 해당하며, 사용자가 글 보기 화면에서 글 수정 버튼을 누를 때 기능이 시작된다.

- ① 사용자가 본인이 쓴 글에서 수정 버튼을 누르거나, 관리자가 다른 사용자의 수정 버튼을 누른다.
- ② 글 내용 수정창이 출력되며, 사용자는 글 내용을 수정할 수 있다.
- ③ 서버에 글 수정 요청을 보내고, true가 반환되면 수정 성공 메시지를 띄운다.
- ④ 사용자가 성공 메시지에서 확인 버튼을 누르면 홈 화면으로 전환된다.
- ⑤ 만약 사용자가 본인이 쓰지 않은 글에서 수정 버튼을 누를 경우, 수정 실패 메시지를 띄운다.

#### 4.2.5 Delete post



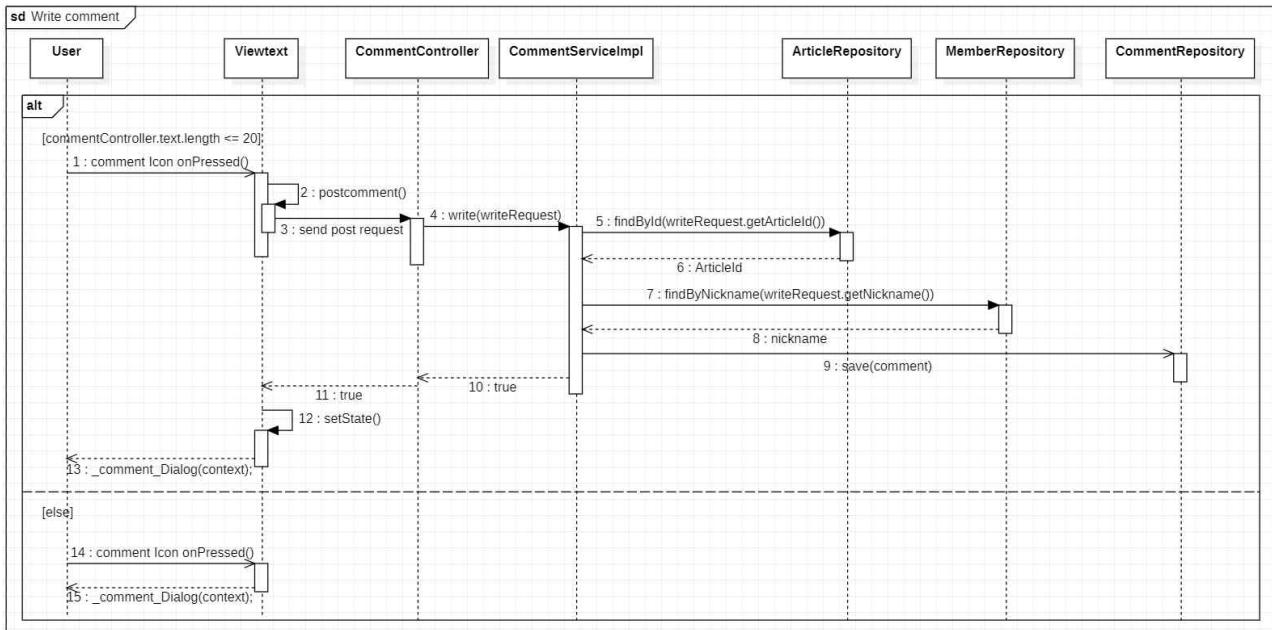
[그림 4-10] Delete post SD

사용자가 글을 삭제하는 use case를 나타내는 sequence diagram이다. <Use case #8>에 해당하며, 사용자가 글 보기 화면에서 글 삭제 버튼을 누를 때 기능이 시작된다.

- ① 사용자가 본인이 쓴 글에서 삭제 버튼을 누르거나, 관리자가 다른 사용자의 삭제 버튼을 누른다.
- ② 서버에 글 삭제 요청을 보내고, true가 반환되면 삭제 성공 메시지를 띄운다.
- ③ 사용자가 성공 메시지에서 확인 버튼을 누르면 홈 화면으로 전환된다.
- ④ 만약 사용자가 본인이 쓰지 않은 글에서 삭제 버튼을 누를 경우, 삭제 실패 메시지를 띄운다.

## 4.3 Comment sequence diagram

### 4.3.1 Write comment

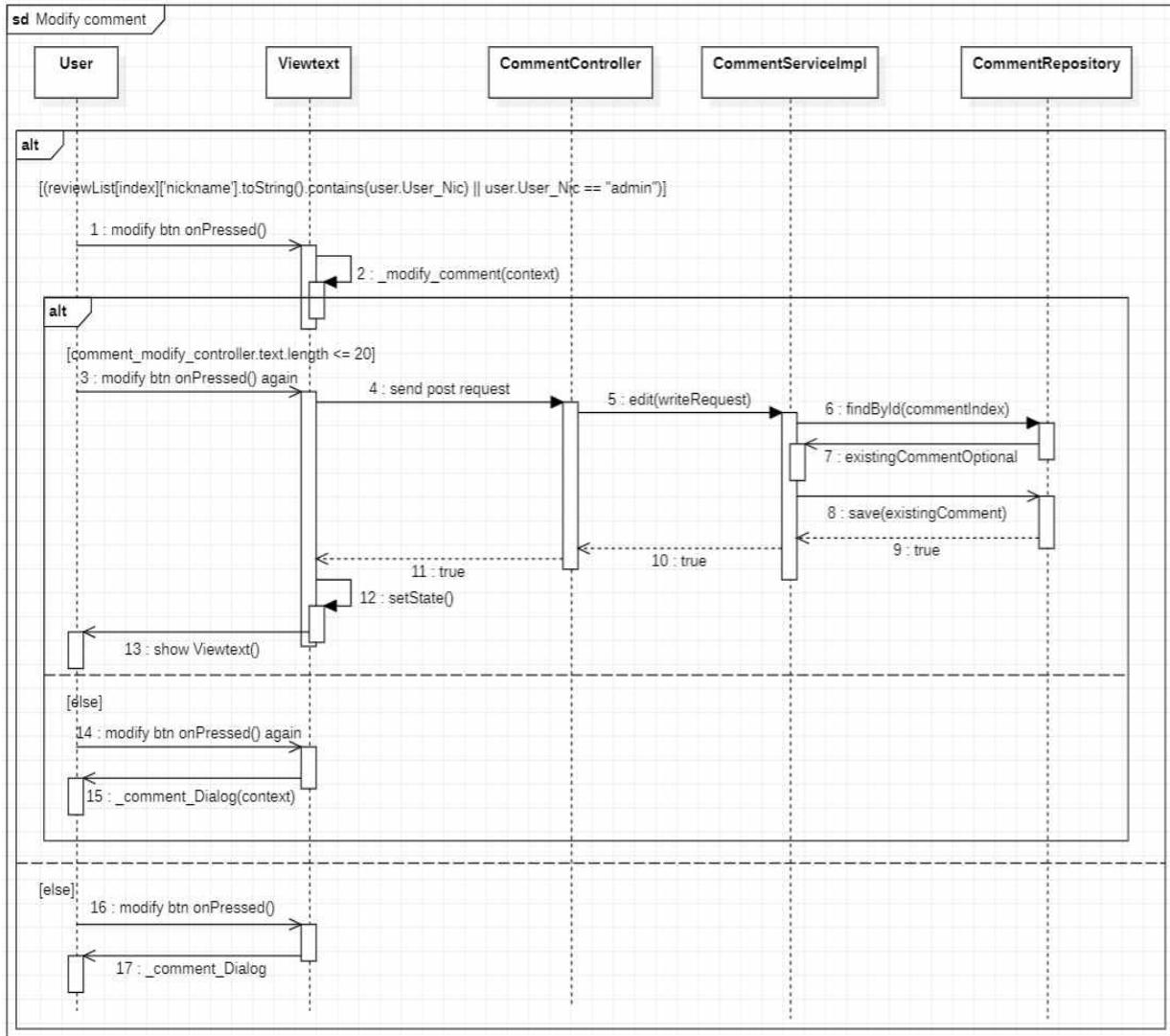


[그림 4-11] Write comment SD

사용자가 댓글을 작성하는 use case를 나타내는 sequence diagram이다. <Use case #9-3>에 해당하며, 사용자가 글 보기 화면 속 댓글 창에서 댓글을 작성하고 댓글 아이콘을 누를 때 기능이 시작된다.

- ① 사용자가 글 보기 화면에서 댓글을 20자 초과하여 입력했을 경우 댓글 작성 실패 메시지를 띄운다.
- ② 사용자가 글 보기 화면에서 댓글을 20자 이내로 입력한 후 댓글 아이콘을 누른다.
- ③ 댓글 작성에 성공하여 서버에 요청을 보낸다. 시스템은 글 번호, 사용자 닉네임을 불러온다.
- ④ 댓글을 서버에 저장하고 true 값을 반환한다.
- ⑤ 댓글 창을 출력하고, 댓글 작성 성공 메시지를 띄운다.

#### 4.3.2 Modify comment

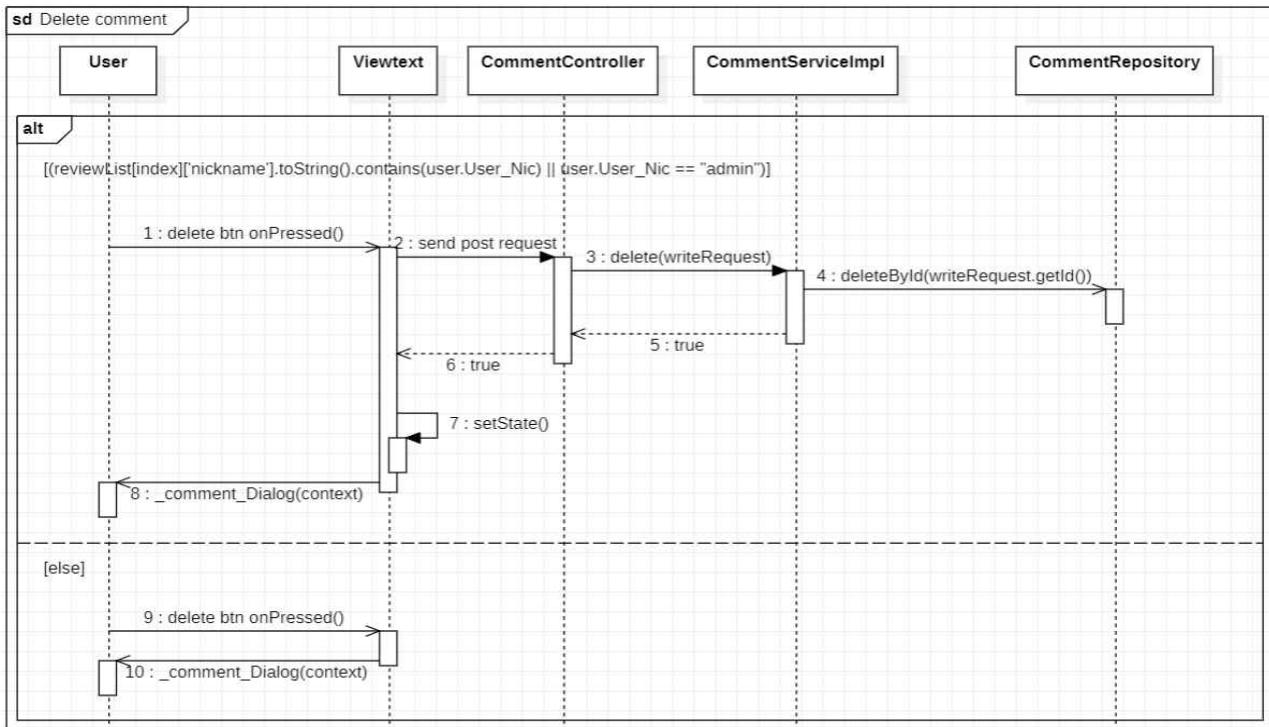


[그림 4-12] Modify comment SD

사용자가 댓글을 수정하는 use case를 나타내는 sequence diagram이다. <Use case #9-5>에 해당하며, 사용자가 글 보기 화면 속 댓글 창에서 댓글을 수정할 때 기능이 시작된다.

- ① 사용자가 자신이 작성한 댓글을 수정하거나, 관리자가 다른 사용자의 댓글을 수정한다. 사용자가 다른 사용자의 댓글을 수정하려고 할 경우 권한이 없다는 에러 메시지를 출력한다.
- ② 사용자나 관리자가 수정한 댓글 길이가 20자 이내일 경우 서버에 요청을 보낸다. 댓글 수정에 성공한 후 글 보기 화면을 출력한다.
- ③ 사용자나 관리자가 수정한 댓글 길이가 20자를 초과할 경우 에러 메시지를 출력한다.

#### 4.3.3 Delete comment

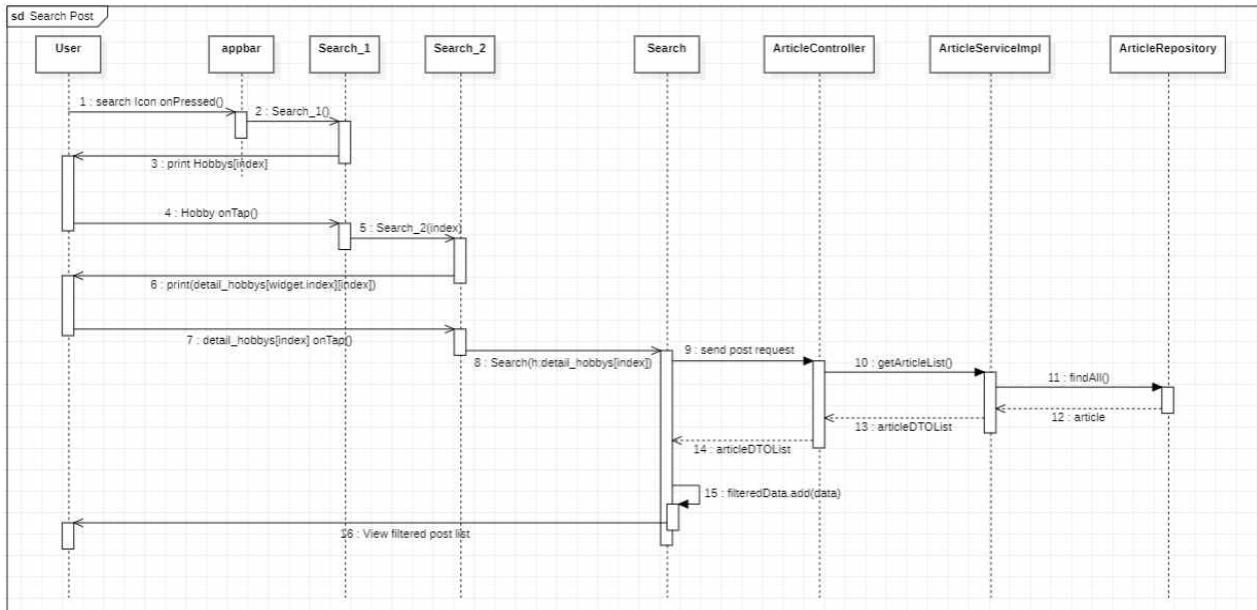


[그림 4-13] Delete comment SD

사용자가 댓글을 삭제하는 use case를 나타내는 sequence diagram이다. <Use case #9-5>에 해당하며, 사용자가 글 보기 화면 속 댓글창에서 댓글을 삭제할 때 기능이 시작된다.

- ① 사용자가 다른 사용자의 댓글을 삭제하려고 할 경우 권한이 없다는 에러 메시지를 출력한다.
- ② 사용자가 자신이 작성한 댓글을 삭제하거나, 관리자가 다른 사용자의 댓글을 삭제할 경우 기능이 시작된다.
- ③ 서버에 댓글 삭제 요청을 보내 댓글을 DB에서 삭제한다.
- ④ 시스템은 삭제에 성공했다는 문구를 띄운다.

## 4.4 Search Post sequence diagram



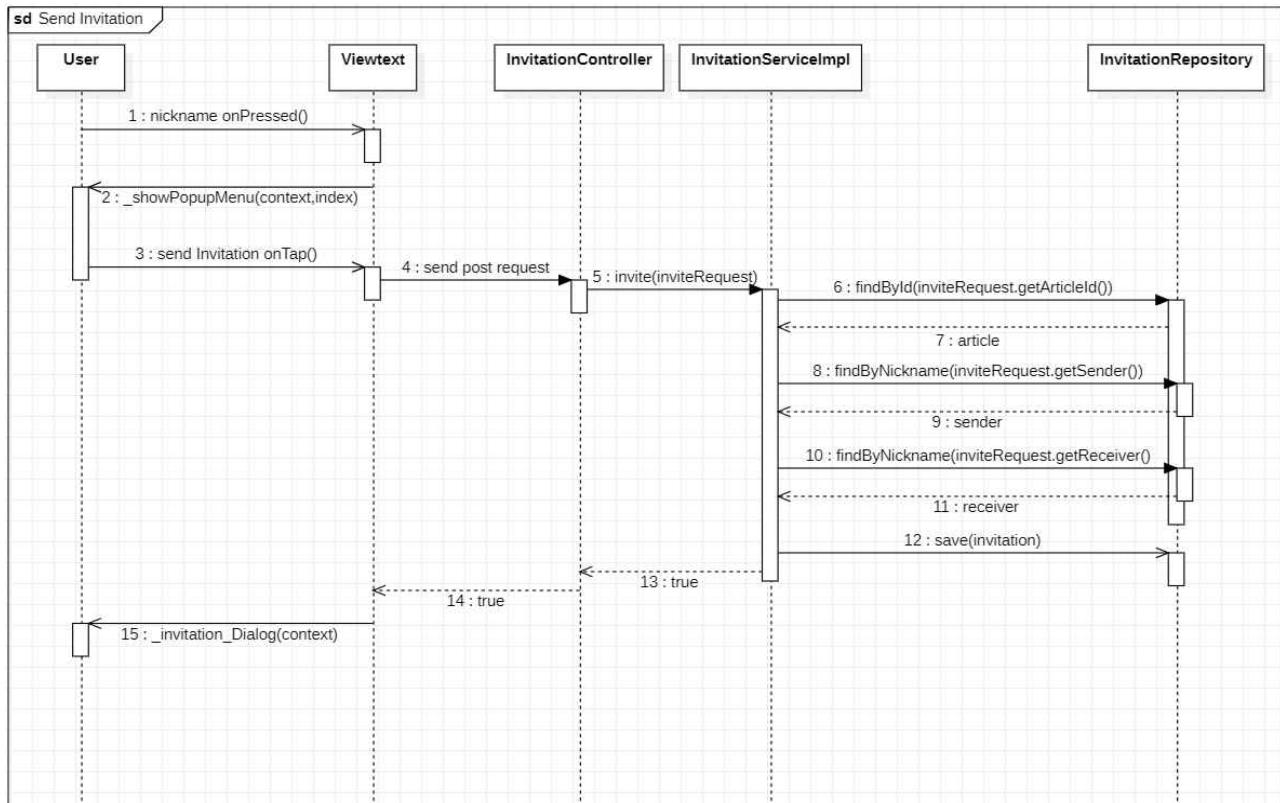
[그림 4-14] Search Post SD

사용자가 원하는 카테고리에 속하는 글을 검색하는 use case를 나타내는 sequence diagram이다. <Use case #10>에 해당하며, 순서는 다음과 같다.

- ① 사용자가 상단 바에 있는 검색 아이콘을 누를 때 기능이 시작된다.
- ② 검색 아이콘을 누르면 시스템은 큰 카테고리 선택 창을 출력한다.
- ③ 사용자가 큰 카테고리 선택 창에서 카테고리를 하나 누르면, 카테고리의 인덱스 번호와 함께 세부 카테고리 선택 창으로 넘어간다.
- ④ 사용자가 세부 카테고리 선택 창에서 세부 카테고리를 하나 누르면, 검색 결과 창이 출력된다.
- ⑤ 서버에 요청하여 전체 글 목록을 불러온다.
- ⑥ 전체 글 목록 중 세부 카테고리에 해당하는 글만 필터링하여 출력한다.

## 4.5 Invitation sequence diagram

### 4.5.1 Send Invitation

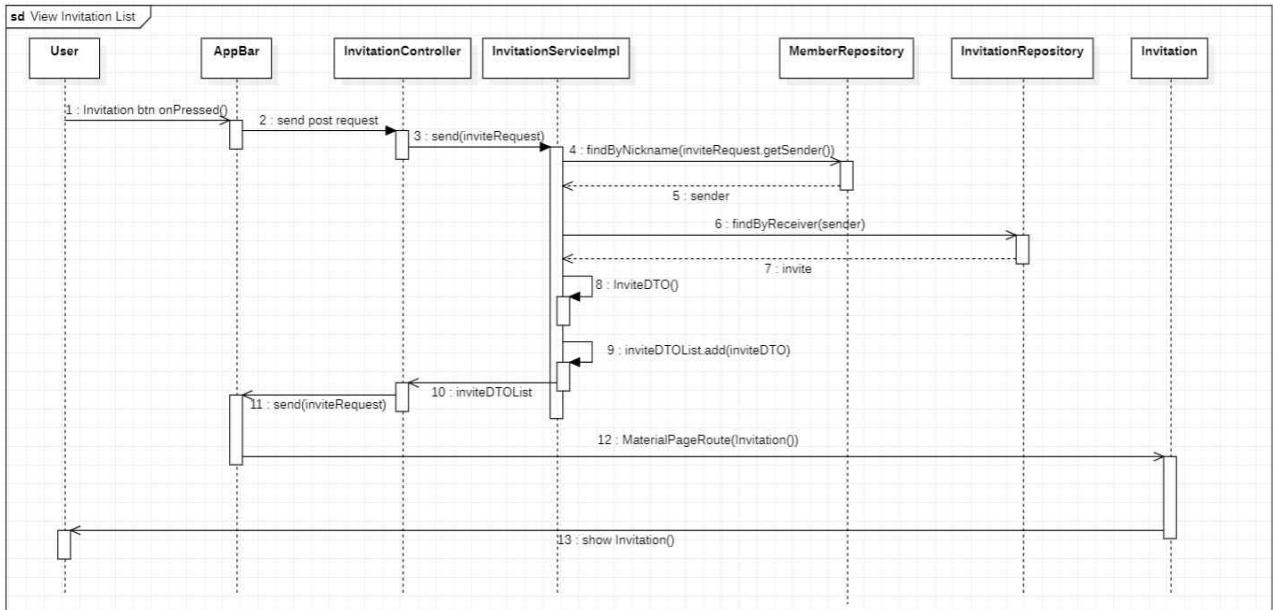


[그림 4-15] Send Invitation SD

사용자가 다른 사용자에게 모임 초대장을 보내는 use case를 나타내는 sequence diagram이다. <Use case #12>에 해당하며, 순서는 다음과 같다.

- ① 사용자가 자신의 글에서 다른 사용자의 닉네임을 눌러 상호작용 창을 열고, 초대장 보내기 버튼을 누른다.
- ② 서버에 요청을 보내 글, 송신자, 수신자를 불러와 초대장을 생성하고 DB에 저장한다.
- ③ 시스템은 초대장 보내기 성공 문구를 출력한다.

#### 4.5.2 View Invitation List

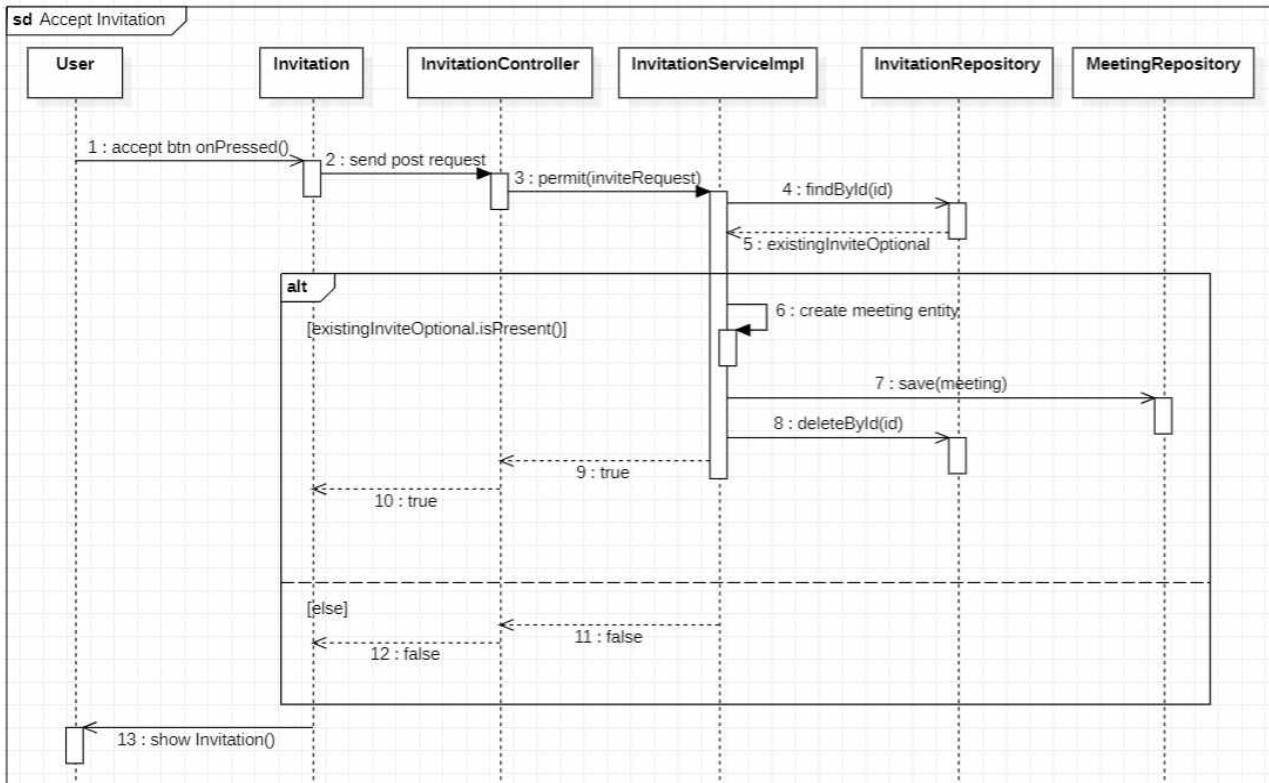


[그림 4-16] View Invitation List SD

사용자가 초대장 목록을 보는 use case를 나타내는 sequence diagram이다. <Use case #13>에 해당하며, 순서는 다음과 같다.

- ① 사용자가 앱 상단 바에 있는 초대장 아이콘 버튼을 누를 때 기능이 시작된다.
- ② 시스템은 서버에 요청을 보내 DB에서 초대장 목록을 가져온다.
- ③ 시스템은 초대장 목록 화면으로 전환한다.
- ④ 사용자는 초대장 목록 화면에서 자신이 받은 초대장 목록을 볼 수 있다.

#### 4.5.3 Accept Invitation

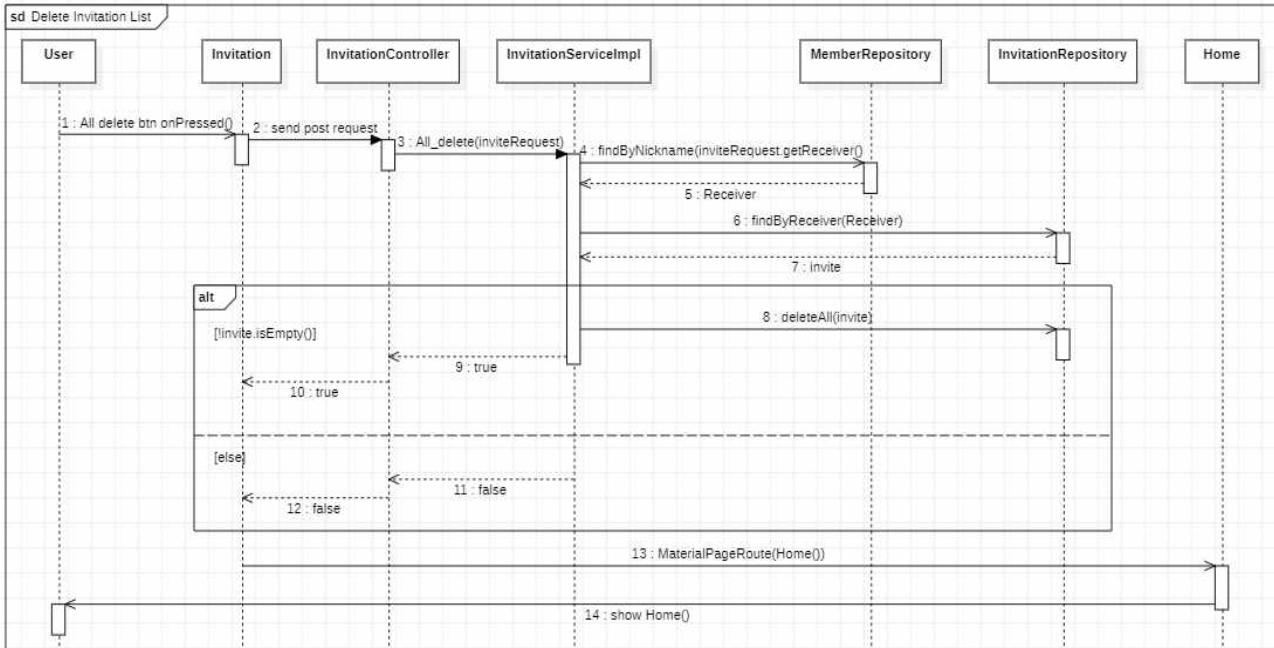


[그림 4-17] Accept Invitation SD

사용자가 댓글을 삭제하는 use case를 나타내는 sequence diagram이다. <Use case #13-5>에 해당하며, 사용자가 초대장을 수락해 일정이 추가되는 과정을 설명한다. 이 SD는 View Invitation List SD에서 이어진다.

- ① 초대장 목록 화면에서 사용자가 초대장을 읽은 후 수락 버튼을 누르면 기능이 시작된다.
- ② 서버에 요청을 보내 DB에서 id를 불러와 초대장 존재 여부를 체크한다.
- ③ 초대장이 있다면 일정을 생성하여 일정 DB에 저장한 후, id를 삭제한다.
- ④ 초대장 목록 화면으로 돌아간다.
- ⑤ 생성된 일정은 View Calendar SD를 통해 볼 수 있다.

#### 4.5.4 Delete Invitation List



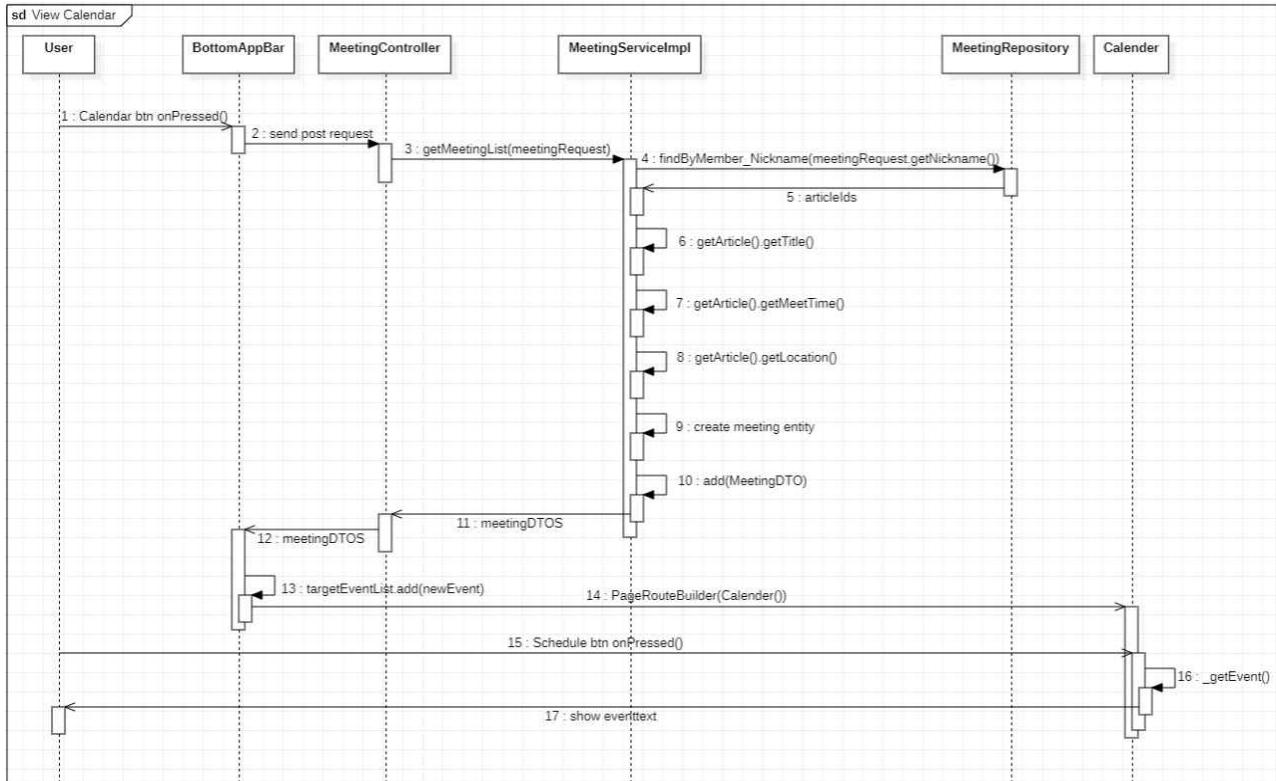
[그림 4-18] Delete Invitation List SD

사용자가 초대장 목록을 모두 삭제하는 use case를 나타내는 sequence diagram이다. <Use case #14>에 해당하며, 이 SD는 View Invitation List SD에서 이어진다.

- ① 사용자가 초대장 목록 화면에서 전체 삭제 버튼을 누른다.
- ② 서버에 요청을 보내 사용자의 닉네임을 불러온 뒤, 초대장 존재 여부를 체크한다.
- ③ 초대장이 있을 경우, 사용자가 받은 초대장을 DB에서 삭제하고 true를 반환한다. 초대장이 없을 경우, false를 반환한다.
- ④ 초대장 목록 화면에서 홈 화면으로 전환한다.

## 4.6 Calendar sequence diagram

### 4.6.1 View Calendar



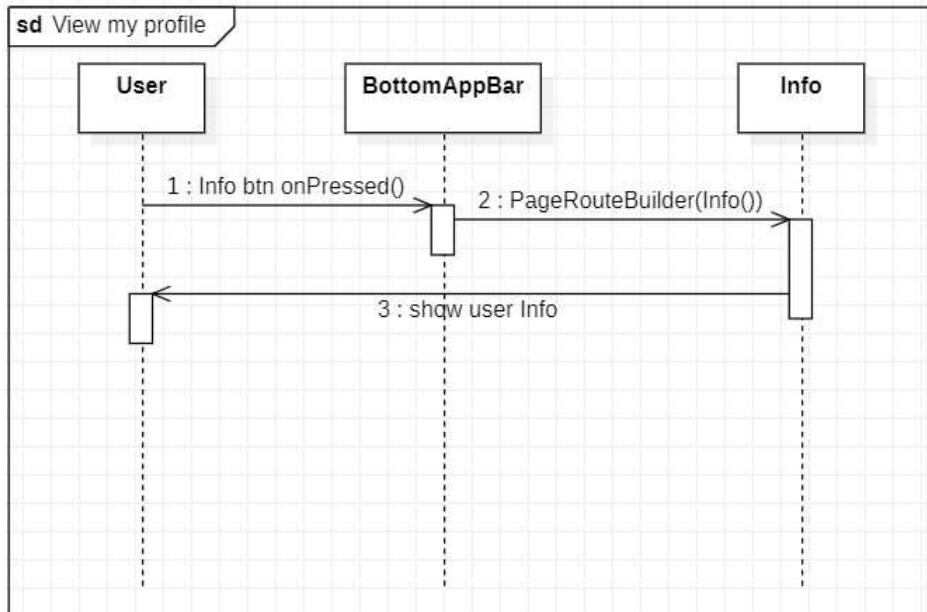
[그림 4-19] View Calendar SD

사용자가 자신의 캘린더 화면을 보는 use case를 나타내는 sequence diagram이다. <Use case #15>에 해당하며, 순서는 다음과 같다.

- ① 사용자가 하단 바에서 일정 버튼을 누를 때 기능이 시작된다.
- ② 서버에 요청을 보내 DB에서 사용자의 일정 목록을 가져와 캘린더 화면에 출력한다.
- ③ 캘린더 화면에서 사용자가 원하는 일정 날짜를 누르면 캘린더 하단에 일정 상세 내용이 출력된다.

## 4.7 Profile sequence diagram

### 4.7.1 View my profile

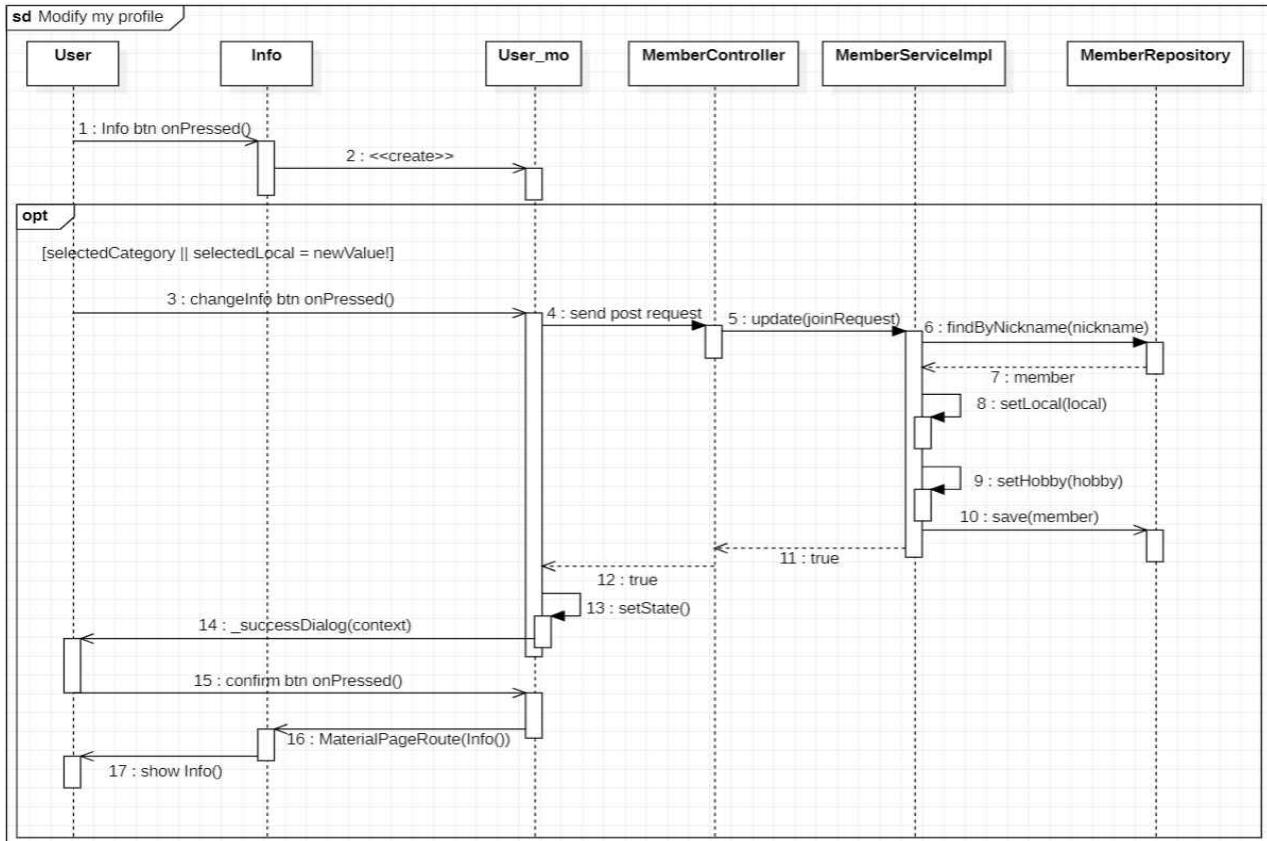


[그림 4-20] View my profile SD

사용자가 내 정보 화면을 보는 use case를 나타내는 sequence diagram이다. <Use case #16-1>에 해당하며, 순서는 다음과 같다.

- ① 사용자가 내 정보 탭을 누르면 내 정보 화면이 출력된다.
- ② 내 정보 화면 속 정보는 로그인 시 서버를 통해 불러오기 때문에 별도의 서버 접근이 필요하지 않다.

#### 4.7.2 Modify my profile

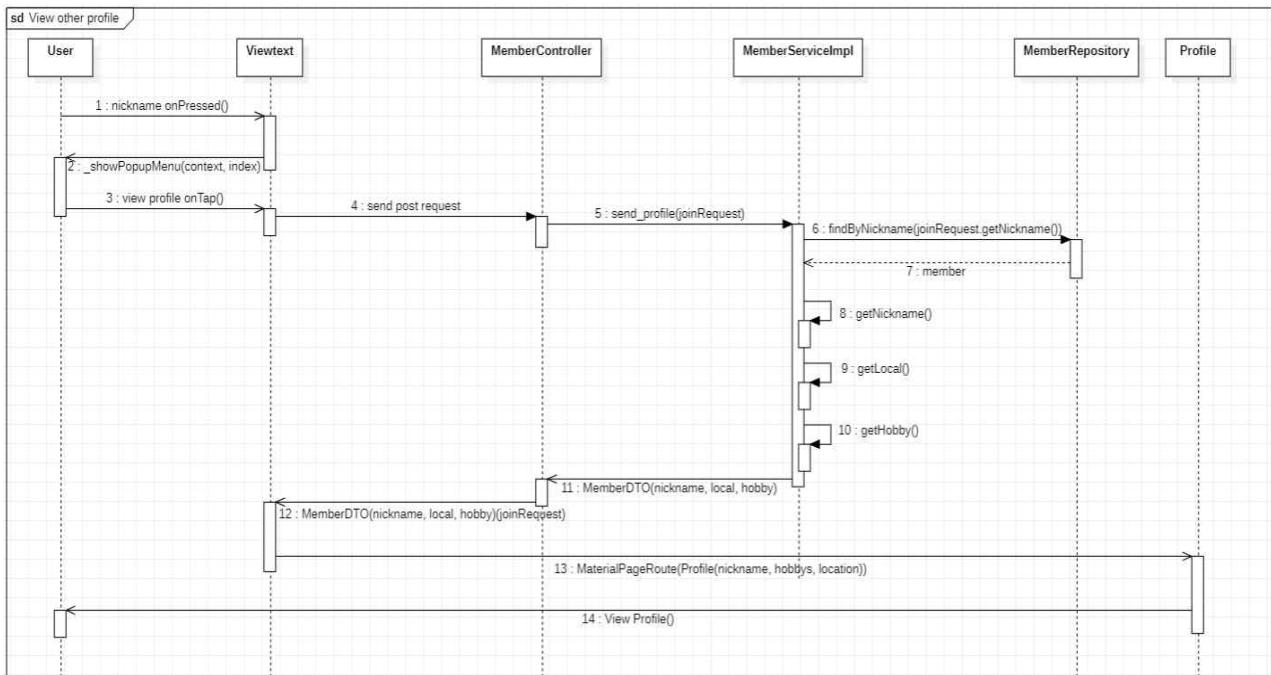


[그림 4-21] Modify my profile SD

사용자가 자신의 정보 내용을 수정하는 use case를 나타내는 sequence diagram이다. <Use case #16> main success scenario 2~7에 해당하며, 내 정보 수정 순서는 다음과 같다.

- ① 사용자가 내 정보 화면에서 회원 정보 수정 버튼을 누를 때 기능이 시작된다. 서버에 요청을 보내 사용자의 아이디와 닉네임 정보를 가져오고, 이를 취미 . 지역 선택 창과 함께 출력한다.
- ② 사용자가 취미나 지역을 다시 선택한 후 회원 수정 버튼을 누른다. 서버에 수정 요청이 가고, 시스템은 사용자가 수정한 부분을 업데이트한다.
- ③ 정보 수정에 성공할 경우, 시스템은 수정 성공 알림 창을 띄운다.
- ④ 사용자가 수정 성공 알림 창에서 확인 버튼을 누를 경우, 내 정보 화면으로 돌아간다.

#### 4.7.3 View other profile



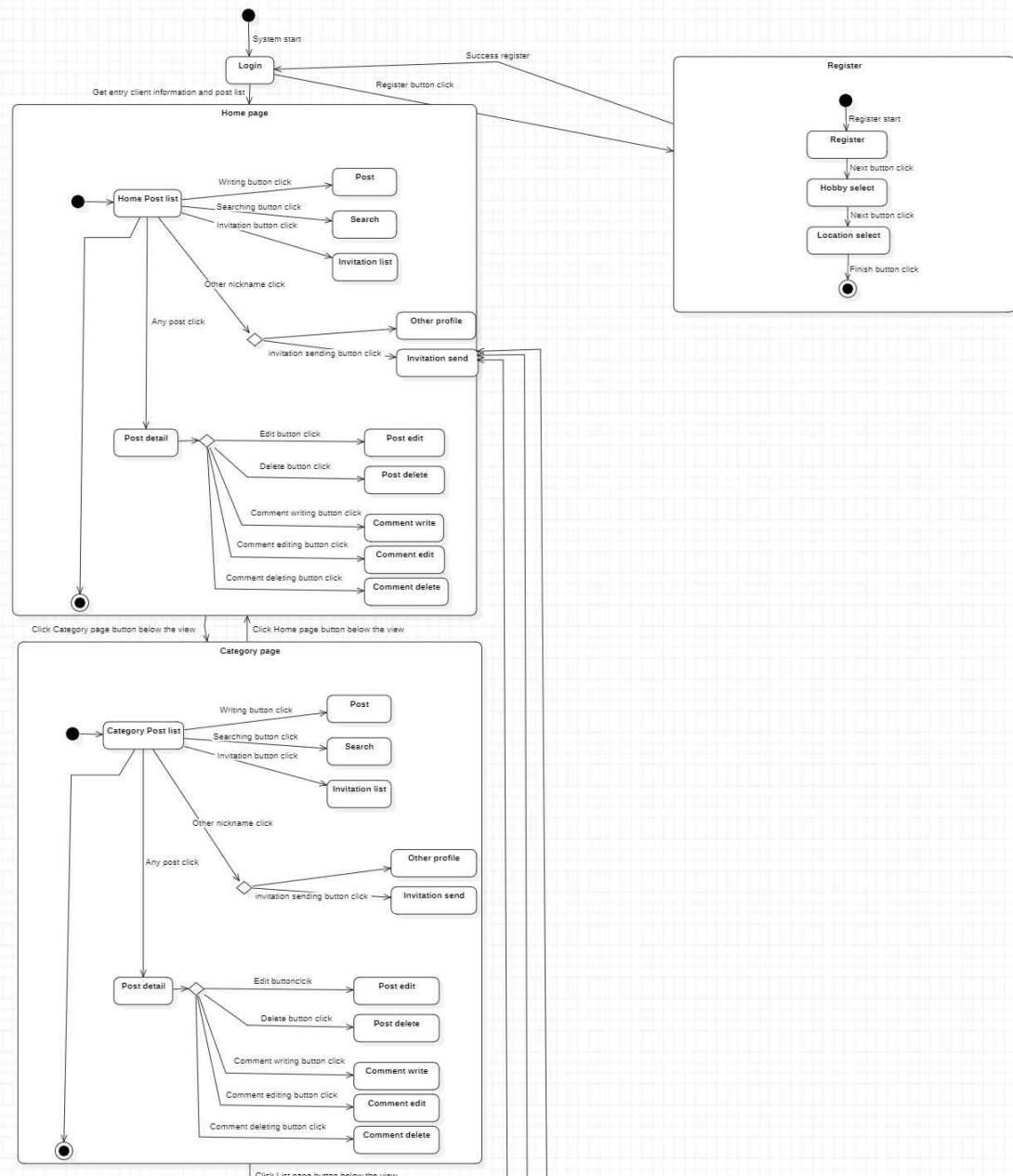
[그림 4-22] View other profile SD

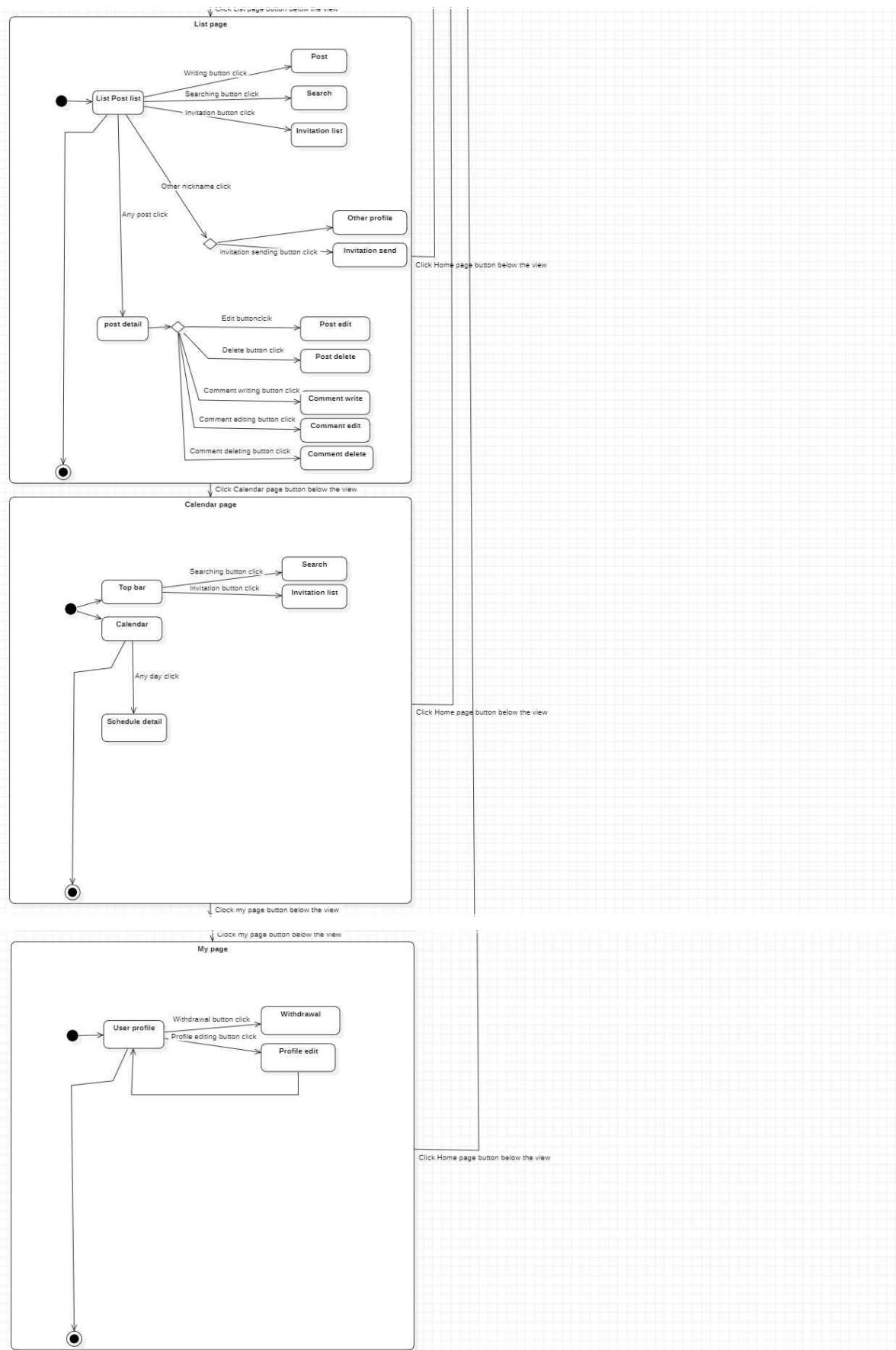
사용자가 다른 사용자의 정보를 확인하는 use case를 나타내는 sequence diagram이다. <Use case #11>에 해당하며, 순서는 다음과 같다.

- ① 사용자는 글 보기 화면에서 다른 사용자의 닉네임을 눌러 상호작용 창을 띄운다.
- ② 사용자가 상호작용 창에서 다른 사용자의 프로필 보기 버튼을 누를 때 기능이 시작된다.
- ③ 서버에 요청을 보내 다른 사용자의 정보(닉네임, 취미, 지역, 가입일)를 불러와 MemberDTO를 생성한다.
- ④ 다른 사용자의 정보 화면으로 전환되고, 다른 사용자의 정보가 출력된다.

## 5. State machine diagram

이 섹션에서는 시스템의 state machine diagram을 그리고 설명한다.

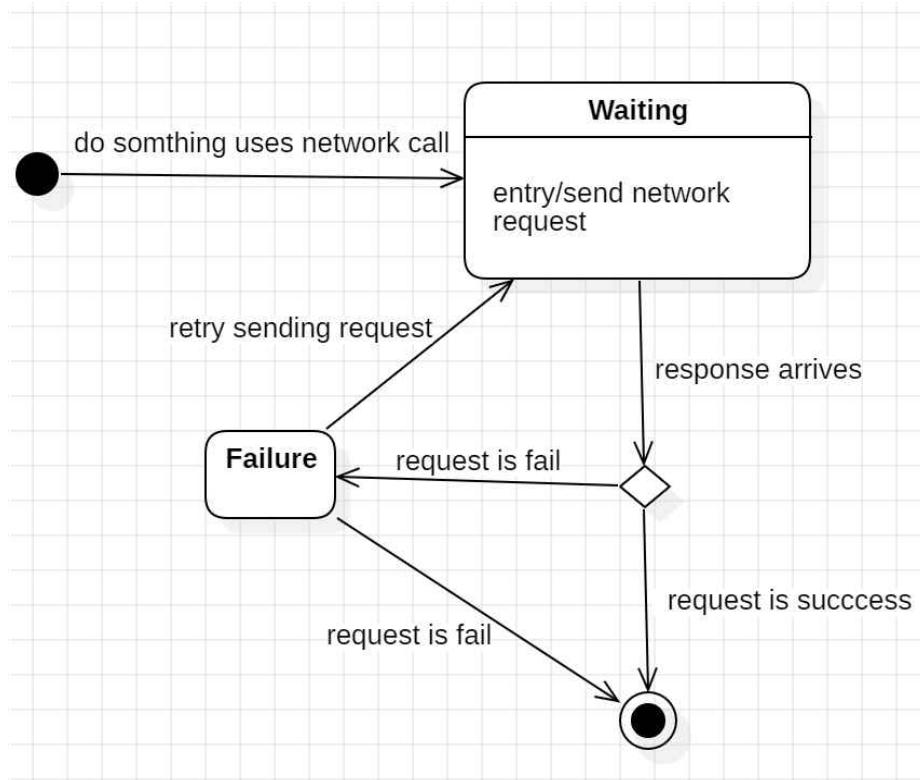




[그림 5-1]

이에 관한 설명은 다음과 같다.

- 각 state는 안드로이드 애플리케이션이 사용자에게 어떠한 화면을 보여주고 있는지에 대한 상태이다. 따라서 state의 이름은 CD의 Fragment 클래스들과 1:1 대응된다.
- 본 SMD에는 5개의 composite state가 있다. 사용자는 시스템을 다른 composite state로 옮기기 위해 화면 하단의 버튼을 누를 수 있다.
- 하단 버튼 클릭은 Dialog로 끝나는 state를 제외한 모든 state에서 실행 가능하다.
- [그림 5-1]은 상황에 따라 state 내부에서 네트워크 요청을 보낼 때가 있는데 이는 공통적으로 network response가 성공적이어야 한다. 요청을 보낼 때의 상태는 [그림 5-2]로 구체화하였다.
- 어떤 때에 네트워크 요청을 보내는지는 SD에서 찾아볼 수 있다.



[그림 5-2]

- State Machine 내의 어떤 한 state가 끝나거나 사용자가 [뒤로가기] 버튼을 누르면 바로 전 화면(state)으로 돌아간다.

## 6. User interface prototype

이 섹션에서는 예상 UI와 UI 안의 각 구성요소를 설명한다.



# Meet Mate

아이디

비밀번호

회원가입

로그인



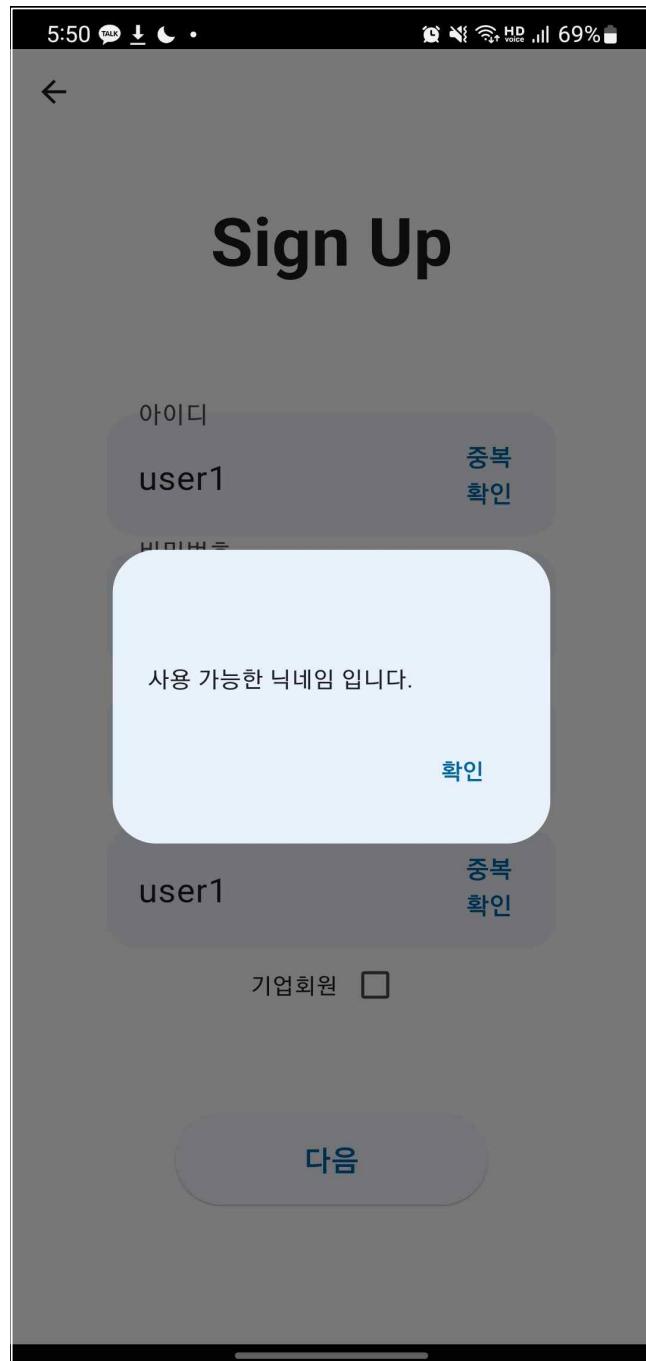
위 그림은 로그인 화면이다. 아이디와 비밀번호를 입력하고 로그인 버튼을 눌러 로그인할 수 있으며, 회원가입 버튼을 클릭하여 아이디와 비밀번호를 생성할 수 있다.



# Sign Up

|  |          |
|--|----------|
| 아이디  | 중복<br>확인 |
| <br>   |          |
| 비밀번호   |          |
| <br>   |          |
| 이름   |          |
| <br>   |          |
| 닉네임 (또는 기업...)   | 중복<br>확인 |
| <br>   |          |
| 기업회원 <input type="checkbox"/>  |          |
| <br>   |          |
| <br><b>다음</b><br><br><div style="width: 60%; margin: auto;"><div style="width: 100%; height: 5px; background-color: black; position: relative;"><div style="width: 10%; height: 100%; background-color: #ccc; position: absolute; left: 0; top: 0;"></div><div style="width: 90%; height: 100%; background-color: #ccc; position: absolute; left: 10%; top: 0;"></div></div></div> |          |

위 그림은 회원가입 화면이다. 아이디와 비밀번호, 이름, 닉네임을 입력하여 회원가입을 진행할 수 있다. 기업 회원의 경우, 체크박스를 이용한다. 다음 버튼을 클릭하여 다음 단계로 이동할 수 있다.



위 그림은 회원가입 과정에서 사용 가능한 닉네임을 입력 시 화면이다. 확인 버튼을 누른 후 다음 버튼을 누를 시 회원가입 다음 단계로 넘어간다.


←     **취미 선택**

축구 야구 농구

다음

|         |      |
|---------|------|
| 운동      | 축구   |
| 아웃도어/여행 | 야구   |
| 언어      | 농구   |
| 봉사활동    | 배구   |
| 댄스/무용   | 수영   |
| 문화/공연   | 테니스  |
| 기타      | 배드민턴 |



위 그림은 회원가입 시 취미를 선택하는 화면이다. 한 사람당 최대 3개의 취미를 선택할 수 있으며, 큰 카테고리와 세부 카테고리로 나누어 진다. 뒤로가기 버튼을 누를 시 이전 화면으로 이동한다.


← 지역 설정

강남구 강동구 강북구

완료

|    |     |
|----|-----|
| 서울 | 강남구 |
| 경기 | 강동구 |
| 부산 | 강북구 |
| 대구 | 강서구 |
| 인천 | 관악구 |
| 광주 | 광진구 |
| 대전 | 구로구 |
| 울산 | 금천구 |
| 세종 | 노원구 |

위 그림은 회원가입 시 지역 구를 선택하는 화면이다. 한 사람당 최대 3개의 지역 구를 선택할 수 있으며, 선택한 지역에 개설되어 있는 모임 목록을 List 카테고리에서 볼 수 있다. 완료 버튼을 누를 시 회원가입이 완료되며 로그인이 가능하다. 뒤로가기 버튼을 누를 시 이전 화면으로 이동한다.



강남구 강동구 강북구

사용자 관리



글쓰기

cndrasfna

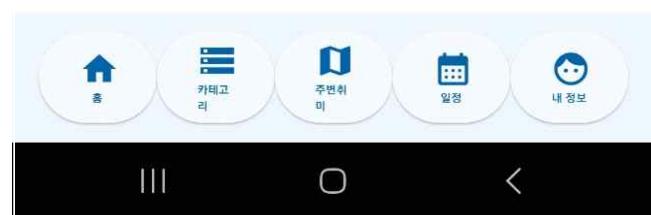
user  
2023-12-01 sdafasdfas...

위 그림은 홈 화면이다. 모든 카테고리의 글 목록을 보여준다. 글 작성이 가능하며 리스트 클릭 시 글 내용을 확인할 수 있다. 하단의 카테고리, 추천취미, 일정, 내 정보 버튼을 통해 다른 탭으로 이동이 가능하다. 상단의 사용자 관리 버튼을 통해 관리자가 사용자를 관리할 수 있다. 상단 우측의 검색 버튼과 초대장 버튼을 통해 검색과 초대장 목록 확인이 가능하다. 상단의 사용자 관리 버튼을 통해 관리자가 사용자를 관리할 수 있다.



강남구 강동구 강북구

사용자 관리

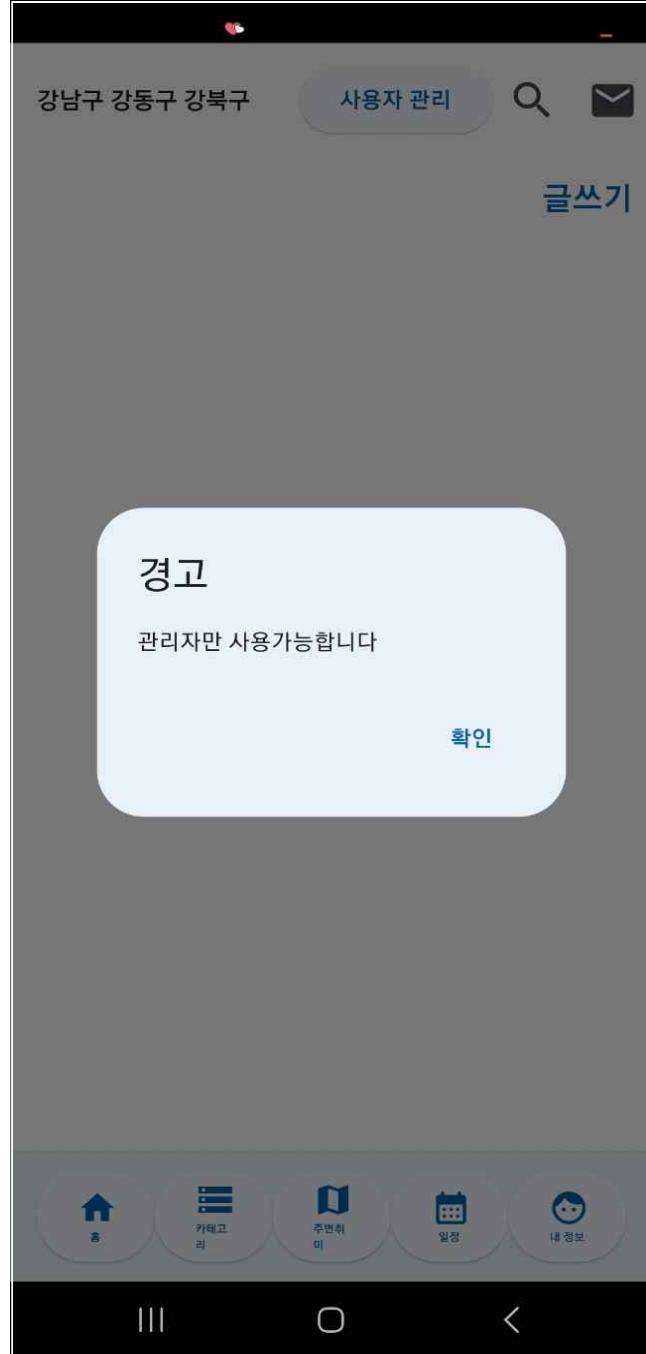


위 그림은 하단의 카테고리 템과 주변취미 템을 눌렀을 시 화면이다. 내 정보에 저장되어 있는 관심취미와 지역설정을 기반으로 그에 해당하는 글 목록만을 확인할 수 있다.

← 관리자 전용 새로고침

|       |   |
|-------|---|
| admin | ⋮ |
| user  | ⋮ |
| user1 | ⋮ |

위 그림은 상단의 사용자 관리 버튼을 눌렀을 때의 화면이다. 관리자는 사용자를 관리할 수 있다.



위 그림은 상단의 사용자 관리 버튼을 일반 사용자가 눌렀을 때의 화면이다. 일반 사용자는 관리자 페이지를 이용할 수 없다.

## ← 글쓰기

제목  
축구

카테고리 운동 ▼ 축구 ▼

지역설정 서울 ▼ 강남구 ▼

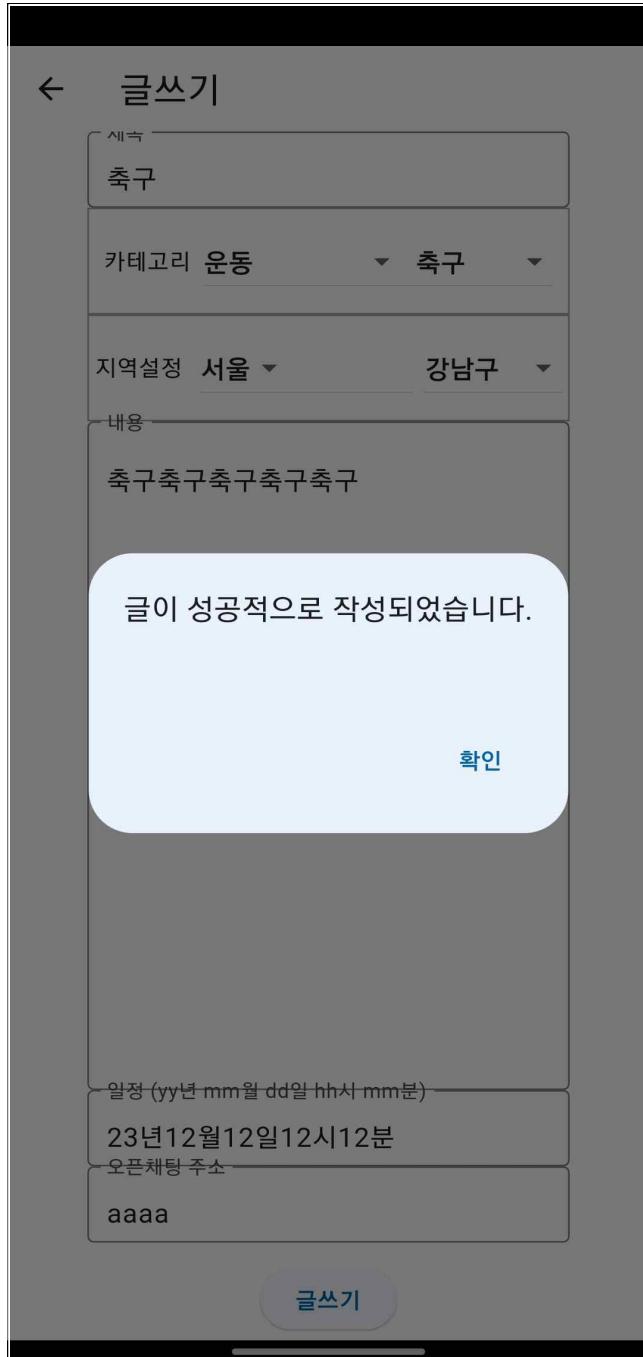
내용  
축구축구축구축구축구

일정 (yy년 mm월 dd일 hh시 mm분)  
23년12월12일12시12분

오픈채팅 주소  
aaaa

글쓰기

위 그림은 글쓰기 화면이다. 제목, 카테고리, 내용 작성 후 글쓰기 버튼을 누르면 해당 카테고리의 글 목록으로 글이 저장된다. 뒤로가기 버튼을 누를 시 이전 화면으로 이동한다.



위 그림은 글이 성공적으로 작성되었을 시 출력되는 화면이다. 확인 버튼을 누를 시 글목록 화면으로 돌아간다.

5:51    •     68% 

← cnbrasfna

user

sdafasdfasdf

강남구 23년 12월 12일 12시 12분

수정

삭제

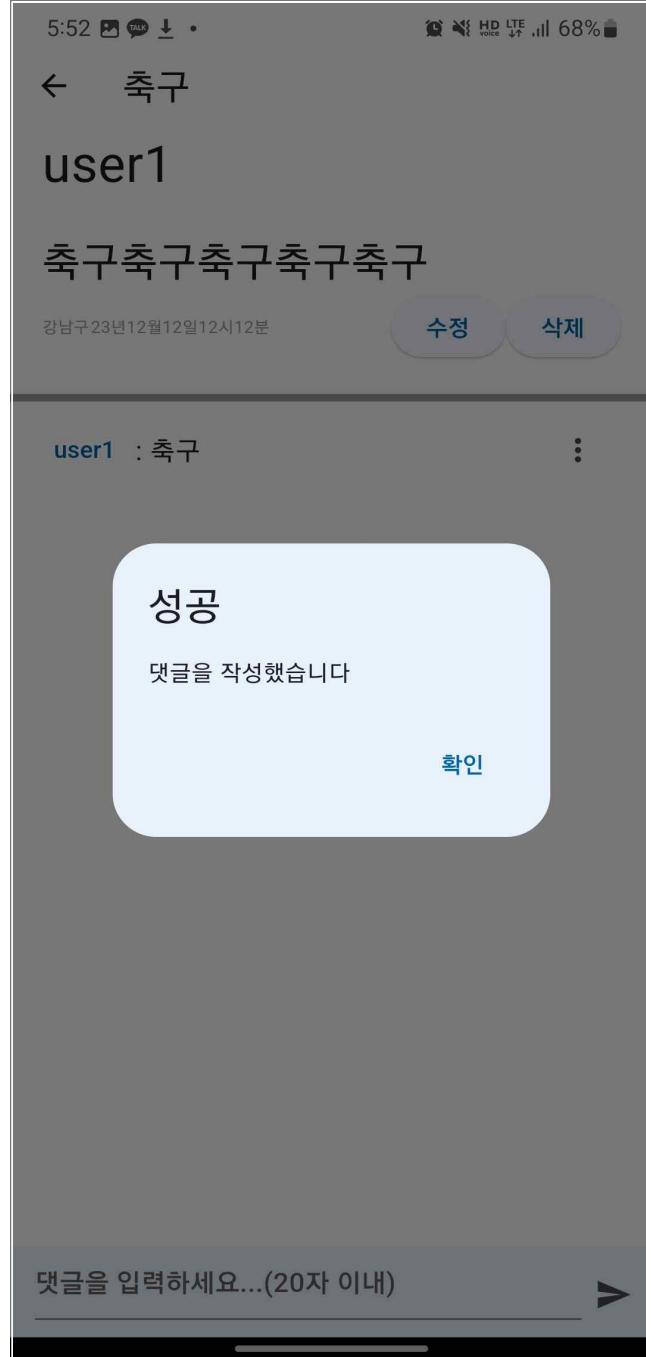
user : dfssdff

⋮

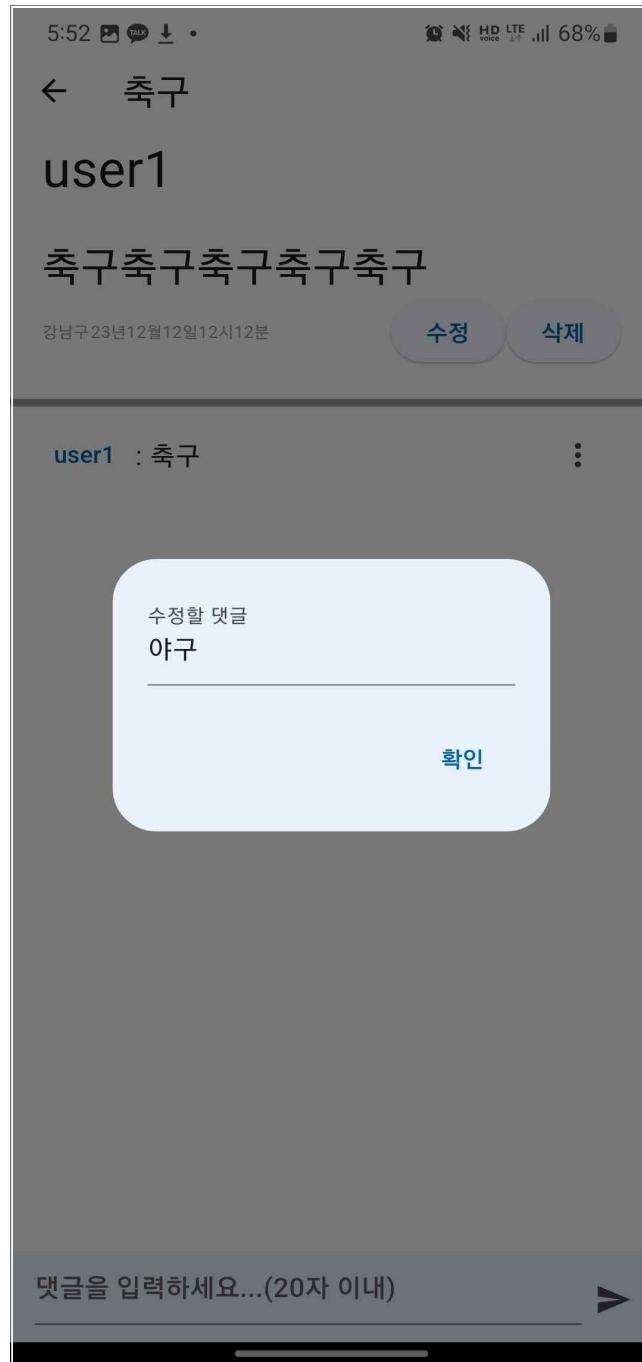
댓글을 입력하세요... (20자 이내)



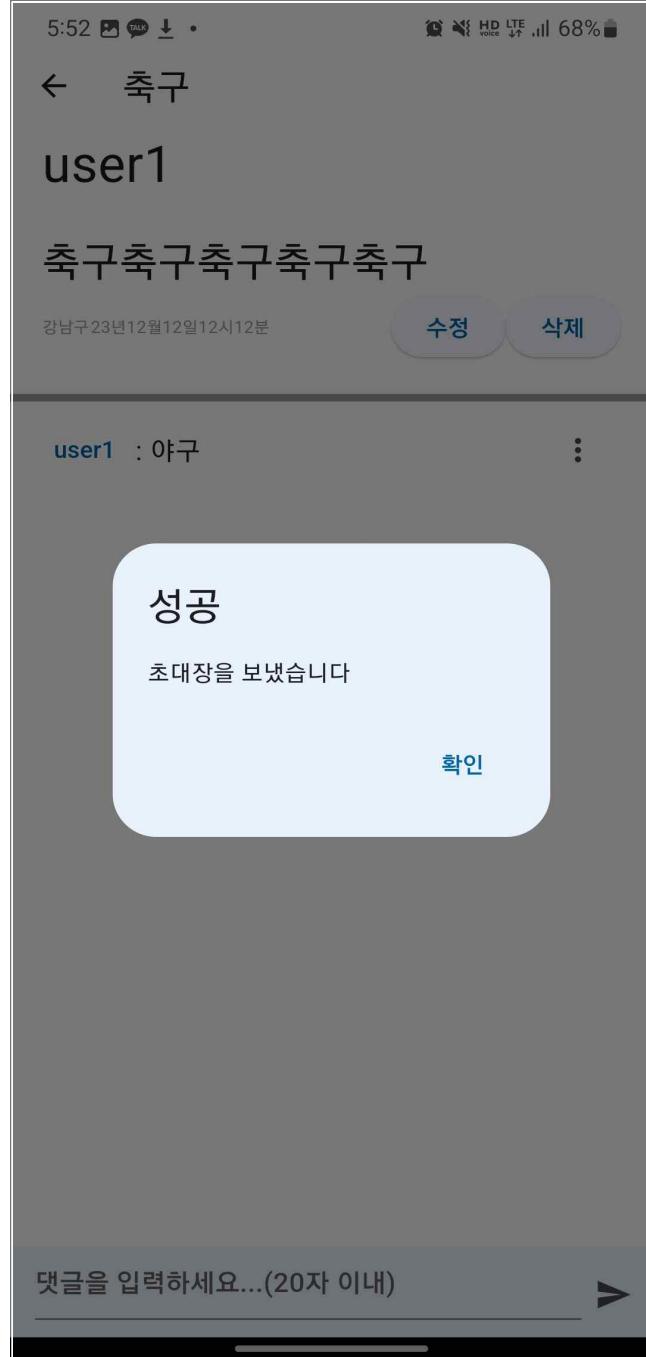
위 그림은 글쓰기에 성공하여 댓글 작성이 가능한 상태의 화면이다. 글의 작성자는 수정과 삭제 버튼을 통해 본인이 쓴 글을 수정, 삭제할 수 있다. 20자 이내의 댓글을 작성 후 올리기 아이콘을 선택하여 댓글 작성을 완료할 수 있다.



위 그림은 댓글 작성에 성공하였을 시 출력되는 화면이다. 확인을 누를 시 글과 댓글이 보이는 화면으로 돌아간다.

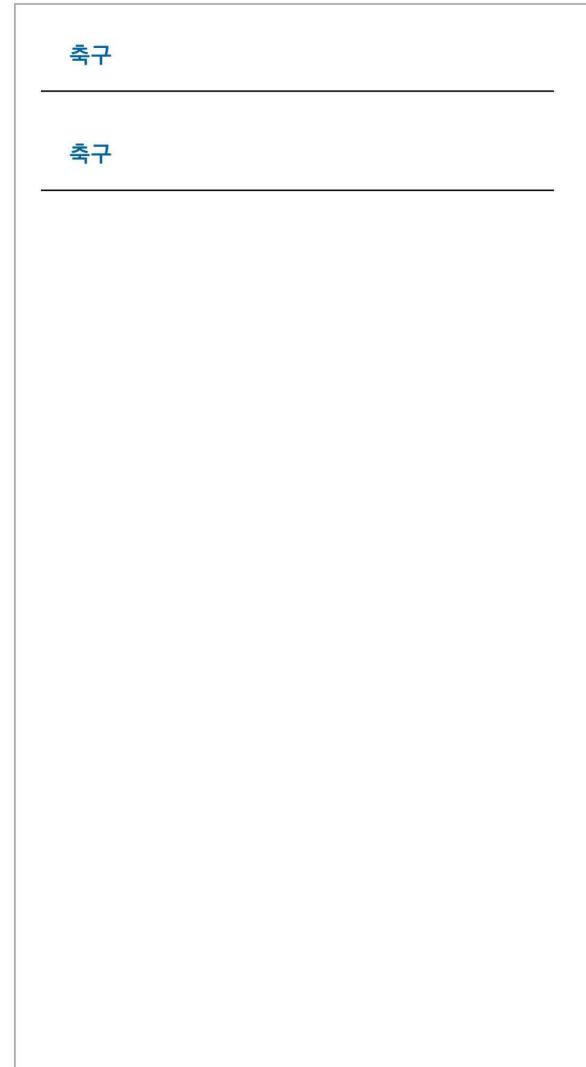


위 그림은 댓글을 수정할 시 출력되는 화면이다. 수정할 댓글 내용을 입력 후 확인 버튼을 누르면 댓글 수정이 가능하다.

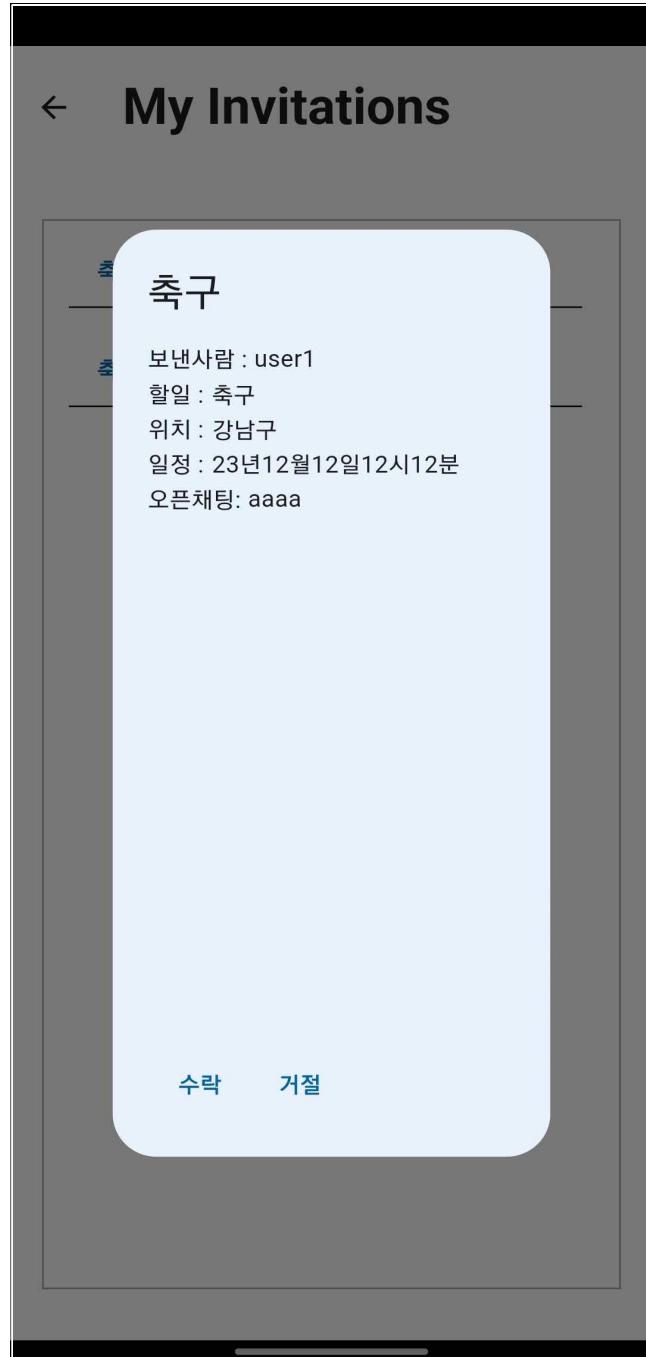


위 그림은 댓글의 상세 선택 버튼을 눌러 초대장 보내기를 클릭하여 성공하였을 경우 출력되는 화면이다.

← My Invitations



위 그림은 상단바의 초대장 버튼을 눌렀을 시 출력되는 화면이다. 초대장 선택 시 상세사항을 확인할 수 있다.



위 그림은 상단바의 초대장 버튼을 누른 후 목록에서 초대장을 선택했을 시 출력되는 화면이다. 수락버튼을 누를 시 캘린더에 일정이 자동 저장되며, 거절버튼을 누를 시 캘린더에 저장되지 않는다.

5:53 TALK • HD voice LTE 68%

← 일정

새로고침

◀ December 2023 ▶

2 weeks

| Sun | Mon | Tue | Wed | Thu | Fri | Sat |
|-----|-----|-----|-----|-----|-----|-----|
| 26  | 27  | 28  | 29  | 30  | 1   | 2   |
| 3   | 4   | 5   | 6   | 7   | 8   | 9   |
| 10  | 11  | 12  | 13  | 14  | 15  | 16  |
| 17  | 18  | 19  | 20  | 21  | 22  | 23  |
| 24  | 25  | 26  | 27  | 28  | 29  | 30  |
| 31  | 1   | 2   | 3   | 4   | 5   | 6   |

시간                      내용                      위치

12시12분                      축구                      강남구



위 그림은 캘린더 화면이다. 자신의 모임 일정을 확인할 수 있다. 뒤로가기 버튼을 누를 시 이전 화면으로 이동한다. 새로고침 버튼을 누를 시 새롭게 추가도니 캘린더 일정을 확인할 수 있다. 사용자는 날짜를 선택하여 일정의 상세사항을 확인할 수 있다.

강남구 강동구 강북구

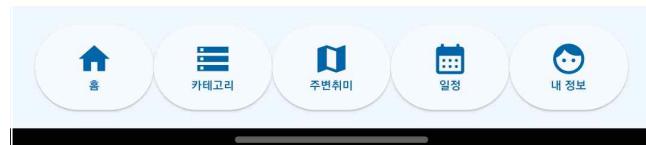
사용자 관리



회원 정보 수정

로그아웃

회원 탈퇴



위 그림은 내 정보 버튼을 눌렀을 시 출력되는 화면이다. 회원 정보 수정, 로그아웃, 회원 탈퇴 버튼을 통해 각 기능을 수행시킬 수 있다.

## ← 회원수정

사용자 ID:user1  
사용자 닉네임 : user1

## 취미선택

운동 ▼ 축구 ▼

운동 ▼ 축구 ▼

운동 ▼ 축구 ▼

## 위치선택

서울 ▼ 강남구 ▼

서울 ▼ 강남구 ▼

서울 ▼ 강남구 ▼

회원수정

위 그림은 회원정보수정 버튼을 눌렀을 시 출력되는 화면이다. 취미와 위치를 수정할 수 있다. 회원수정 버튼을 통해 수정한 내용을 저장시킬 수 있다.



위 그림은 로그아웃 버튼을 눌렀을 시 화면이다. 네 버튼을 누를 시 로그아웃된다. 아오를 누를 시 이전 화면인 내 정보 페이지로 돌아간다.



위 그림은 회원 탈퇴 버튼을 누를 시 화면이다. 네 버튼을 누를 시 회원 탈퇴된다. 아니오 버튼을 누를 시 이전 화면인 내 정보 페이지로 돌아간다.

## 7. Implementation requirements

### ① Application

To install and run the apk file,

- Processor (CPU): Dual core of 1.5 GHz or faster recommended.
- Memory (RAM): 2GB or more recommended.
- Storage: 30MB or more of available internal storage.
- Device firmware : Compatible with Android 13 or higher.

### ② Server

To run the jar file,

- Processor (CPU): Dual core of 2.0 GHz or faster recommended.
- Memory (RAM): 4GB or more recommended.
- Storage space: 20GB or more of available hard disk space.
- OS : Windows 10 or higher, macOS 10.12 or higher, or Linux operating system.
- Java Runtime : Java Runtime Environment (JRE) 8 or higher.

## 8. Glossary

| Terms    | Description  |
|----------|--|
| home     | 취미 카테고리, 지역구 상관없이 모든 글을 볼 수 있는 탭이다. use case diagram의 view entire post에 해당한다.            |
| category | 사용자가 회원가입 때 선택했던 취미 카테고리에 속하는 글을 볼 수 있는 탭이다. use case diagram의 view category post에 해당한다. |
| list     | 사용자가 회원가입 때 선택했던 지역구에 속하는 글을 볼 수 있는 탭이다. use case diagram의 view area post에 해당한다.         |
| 상호작용창    | 다른 사용자의 닉네임을 눌렀을 때 화면 중앙에 뜨는 창이다. '프로필 보기'와 '초대장 보내기' 중에서 하나를 선택할 수 있다.                  |
| 모임       | 사용자가 시스템을 통해 글을 작성하고 초대장을 보내어 자신의 취미나 관심사와 관련된 활동에 참여하고자 하는 사람들을 모집한다.                   |

## 9. References

- 1) 소프트웨어설계 수업 자료 [SWDesign] Use case diagram I, II
- 2) <https://docs.staruml.io/working-with-uml-diagrams/sequence-diagram>
- 3) 소프트웨어설계 수업 자료 [SWDesign] Behavior modeling I, II