

## 2. SDS (Software Design Specification)

## BOOK &amp; Y

| 번호 | 학번       | 이름  |
|----|----------|-----|
| 1  | 22113654 | 송유진 |
| 2  | 22012156 | 박규민 |
| 3  | 22212052 | 최효정 |
| 4  | 22113653 | 정승환 |
| 5  | 22012071 | 방민승 |

## [ Revision history ]

| Revision date | Version # | Description      | Author |
|---------------|-----------|------------------|--------|
| 08/11/2024    | 0.00      | first draft      | -      |
| 06/12/2024    | 0.10      | 전자책 뷰어, 그룹 기능 추가 | -      |
|               |           |                  |        |
|               |           |                  |        |
|               |           |                  |        |
|               |           |                  |        |

## = Contents =

|     |                                   |
|-----|-----------------------------------|
| 1.  | Introduction .....                |
| 2.  | Use case analysis .....           |
| 3.  | Class diagram .....               |
| -   |                                   |
| 3.1 | DB관점 클래스다이어그램 .....               |
| 3.2 | 로그인◦회원가입 클래스다이어그램 .....           |
| 3.3 | 구매 클래스다이어그램 .....                 |
| 3.4 | 그룹가입 클래스다이어그램 .....               |
| 3.5 | 그룹플랜 클래스다이어그램 .....               |
| 4.  | Sequence diagram .....            |
| 5.  | State machine diagram .....       |
| 6.  | User interface prototype .....    |
| 7.  | Implementation requirements ..... |
| 8.  | Glossary .....                    |
| 9.  | References .....                  |

= Authors for each section =

Introduction – 송유진

Use case analysis – 정승환

Class diagram – 최효정, 박규민

Sequence diagram – 송유진

State machine diagram – 송유진

User interface prototype - 방민승

Implementation requirements - 최효정, 박규민

Glossary - 최효정

References - 정승환, 최효정, 박규민, 방민승, 송유진

## 1. Introduction

Summarize the contents of this document.  
– Describe the important points of your design.  
12pt, 160%.

본 문서는 우리 팀에서 개발하고자 하는 웹 사이트인 'Book&y'의 **software design specification**이다. 본 문서에서는 요구사항 분석에서 명시된 사항을 여러가지 관점에서 구체화하고 분명히하여 프로젝트의 전체적인 흐름과 동작을 이해할 수 있도록 작성하였다.

본 문서를 작성할 때 중요하게 생각한 점이 동작, 데이터의 일관성이다. 그래서 다이어그램을 작성할 때 **usecase->sequence->state->class->db** 순으로 계층적으로 작성하였다. 즉, 그 전단계에서 필요한 요구사항(기능, 데이터)을 확인하고 같은 기능, 데이터를 사용할 수 있도록 일치시켰다. 각 다이어그램마다 누락된 데이터, 기능이 없는지 주의깊게 작성하였다.

본 프로젝트는 웹에서 동작하는 소프트웨어로써 **sequence** 다이어그램에서 웹과 서버간의 통신을 자세히 기술하였으며 **class diagram**과 **state machine diagram**을 통해 전반적인 시스템 구조에 대해 알 수 있다. 또한 **user interface prototype**을 통해 소프트웨어에서 실현하고자 하는 동작들을 시각화하였다. 이는 모두 **use case analysis**에 기반하여 작성하였으며 모든 다이어그램의 동작과 일치한다. 또 **implementation requirements**를 통해 소프트웨어가 실행되기 위한 조건에 대해서도 명시하였다.

## 2. Use case analysis

이번 장에서는 Use case diagram, Use case summary, Use case description을 제공한다.

Use case diagram은 description에 대한 부가적인 결과물에 불과하다. 따라서 본 문서를 읽는 독자는 Use case diagram보다 Use case description에서 Use cases에 대한 충분한 정보를 얻어야 한다.



## 2.1 Use case summary

| Use Case Name          | Use Case ID | Title      | Actor  |
|------------------------|-------------|------------|--------|
| <b>mainpage</b>        | #1          | 메인페이지      | User   |
| <b>mypage</b>          | #2          | 마이페이지      | Member |
| <b>register</b>        | #3          | 회원가입       | User   |
| <b>login</b>           | #4          | 로그인        | User   |
| <b>User_image</b>      | #5          | 프로필 이미지 변경 | Member |
| <b>User_nickname</b>   | #6          | 닉네임 수정     | Member |
| <b>setPwd</b>          | #7          | 비밀번호 변경    | Member |
| <b>myHistory</b>       | #8          | 나의 활동내역    | Member |
| <b>purchaseHistory</b> | #9          | 나의 구매/결제내역 | Member |
| <b>User_group</b>      | #10         | 나의 활동그룹    | Member |
| <b>Support</b>         | #11         | Q&A        | User   |
| <b>home</b>            | #12         | 메인화면으로 이동  | User   |

|                        |     |                 |        |
|------------------------|-----|-----------------|--------|
| <b>Log_out</b>         | #13 | 로그아웃            | Member |
| <b>Classification</b>  | #14 | 도서 분류           | User   |
| <b>Bookinfo</b>        | #15 | 책 상세 보기         | User   |
| <b>Detail_Bookinfo</b> | #16 | 책 대여 및 구매희망 페이지 | Member |
| <b>Subscribe</b>       | #17 | 정기대여권           | Member |
| <b>Shopping_cart</b>   | #18 | 장바구니            | Member |
| <b>Purchase_page</b>   | #19 | 구매 페이지          | Member |
| <b>FileViewer</b>      | #20 | 전자책 뷰어          | Member |
| <b>Group List</b>      | #21 | 그룹 목록           | User   |
| <b>Group Info</b>      | #22 | 그룹 소개           | User   |
| <b>Group Creation</b>  | #23 | 그룹 생성           | Member |
| <b>Group Join</b>      | #24 | 그룹 가입           | Member |
| <b>Group Manage</b>    | #25 | 그룹 관리           | Member |
| <b>Group Plan</b>      | #26 | 그룹 플랜           | Member |
| <b>Plan Content</b>    | #27 | 독후감 작성          | Member |

## 2.2 Use case description

| Use Case #1 : mainpage  |  |
|-------------------------|--|
| General Characteristics |  |
| Summary                 | 이 홈페이지의 기본 화면을 담당하는 메인 페이지이다.                            |
| Scope                   |  |
| Level                   | User Level   |
| Author                  |  |
| Last Update             |  |
| Status                  | Analysis   |
| Primary Actor           | User   |
| Preconditions           | 네트워크에 연결되어 있어야 한다.                                       |
| Trigger                 | 웹 사이트에 접속한다.   |
| Success Post Condition  |  |
| Failed Post Condition   |  |
| Main Success Scenario   |  |
| Step                    | Action   |
| 1                       | 이 Use case는 User가 메인페이지에 접근하기 위해 시도한다.                   |
| 2                       | 웹 소프트웨어를 실행한다.   |
| Extension Scenario      |  |
| Step                    | Branching Action   |
| 3                       | 3a. 네트워크 연결에 오류가 발생한다.<br>3a1. 네트워크 연결에 실패했다는 메시지를 출력한다. |
| Related Information     |  |
| Performance             | < 3 second   |
| Frequency               | None   |
| Concurrency             | None   |
| Due Date                | None   |

## Use Case #2 : mypage

### General Characteristics

|                        |  |
|------------------------|--|
| Summary                | 이 홈페이지를 이용하는 사용자들 본인의 개인정보를 확인하거나 수정할 수 있는 화면이다.                       |
| Scope                  |  |
| Level                  | User Level   |
| Author                 |  |
| Last Update            |  |
| Status                 | Analysis   |
| Primary Actor          | User   |
| Preconditions          | 사용자는 로그인이 되어 있어야 한다.<br>네트워크에 연결되어 있어야 한다.                             |
| Trigger                | 사용자가 마이페이지로 이동하는 아이콘을 클릭한다   |
| Success Post Condition | 사용자가 입력한 정보가 Server에 저장되어 있는 회원정보랑 일치하여 로그인에 성공한 상태여야 한다.              |
| Failed Post Condition  | 사용자가 입력한 정보가 Server에 저장되어 있는 회원정보랑 일치하지 않아 로그인에 성공한 상태가 아니라면 이용할 수 없다. |

### Main Success Scenario

| Step | Action  |
|------|---|
| 1    | 이 Use case는 User가 마이페이지에 접근하기 위해 시도한다.                          |
| 2    | 회원 가입때 기입 했던 ID와 Password를 토대로 로그인을 한 후, 마이페이지로 이동하는 아이콘을 클릭한다. |
| 3    | 시스템은 사용자가 로그인에 성공한 상태라면, 마이페이지로 이동할 수 있다.                       |
| 4    | 마이페이지로 이동하면 제공하는 기능을 사용할 수 있다.                                  |

### Extension Scenario

| Step | Branching Action   |
|------|--|
| 3    | 3a. 사용자가 로그인이 되어있지 않다면 마이페이지 이동에 실패한다.<br>3a1. 마이페이지 이동 실패 메시지를 띄운다.<br>3a2. 로그인 페이지로 넘어가게 된다.<br>3b. 네트워크 연결에 오류가 발생한다.<br>3b1. 네트워크 연결에 실패했다는 메시지를 출력한다. |

| Related Information |            |
|---------------------|------------|
| Performance         | < 3 second |
| Frequency           | None       |
| Concurrency         | None       |
| Due Date            | None       |

| Use Case #3 : Register  |  |
|-------------------------|--|
| General Characteristics |  |
| Summary                 | 사용자가 주요 기능을 사용하기 위해 로그인을 하기전 회원가입을 하기 위한 페이지이다.. |
| Scope                   |  |
| Level                   | User Level                                       |
| Author                  |  |
| Last Update             |  |
| Status                  | Analysis   |
| Primary Actor           | User   |
| Preconditions           | 네트워크에 연결되어 있어야 한다.                               |
| Trigger                 | 사용자가 로그인 페이지에서 회원가입 배너를 클릭한다.                    |
| Success Post Condition  |  |
| Failed Post Condition   |  |
| Main Success Scenario   |  |
| Step                    | Action   |
| 1                       | 0) Use case는 User가 회원가입하기 위해 접근한다.               |
| 2                       | 로그인 페이지에서 회원가입 배너를 클릭한다.                         |
| 3                       | 회원가입 페이지로 이동하면, 양식에 맞게 정보를 기입한다.                 |
| 4                       | 회원 가입에 성공한다.                                     |
| Extension Scenario      |  |

| Step                | Branching Action  |
|---------------------|---|
| 3                   | 3a. 사용자가 잘못된 정보나 중복된 id를 입력한다.<br>3a1. 사용자가 잘못된 정보를 입력했거나 아이디가 유효하면 회원<br>가입에 실패한다.<br>3b. 네트워크 연결에 오류가 발생한다.<br>3b1. 네트워크 연결에 실패했다는 메시지를 출력한다. |
| Related Information |   |
| Performance         | < 3 second  |
| Frequency           | None  |
| Concurrency         | None  |
| Due Date            | None  |

## Use Case #4 : Login

### General Characteristics

|                        |   |
|------------------------|---|
| Summary                | 이 홈페이지의 여러가지 기능을 사용하기 위해 로그인을 하기 위한 페이지이다.            |
| Scope                  |   |
| Level                  | User Level  |
| Author                 |   |
| Last Update            |   |
| Status                 | Analysis  |
| Primary Actor          | User  |
| Preconditions          | 네트워크에 연결되어 있어야 한다.                                    |
| Trigger                | 사용자가 회원가입때 기입했던 정보를 토대로 Textbox에 ID와 PW를 입력한다.        |
| Success Post Condition | 사용자가 입력한 정보가 Server에 저장되어 있는 회원정보랑 일치하여 로그인에 성공한다.    |
| Failed Post Condition  | 사용자가 입력한 정보가 Server에 저장되어 있는 회원정보랑 일치하지 않다면 이용할 수 없다. |

### Main Success Scenario

| Step | Action   |
|------|--|
| 1    | 이 Use case는 User가 로그인하기 위해 사용한다.                   |
| 2    | 회원 가입때 기입 했던 ID와 Password를 토대로 기입한다.               |
| 3    | 사용자가 로그인에 성공한다면 로그인 상태를 요구로 하는 여러가지 기능들을 사용할 수 있다. |

### Extension Scenario

| Step | Branching Action  |
|------|---|
| 3    | 3a. 사용자가 회원가입 시 기입했던 정보와 일치하지 않는 정보를 기입한다.<br>3a1. 로그인 실패 메시지를 띄운다.<br>3b. 네트워크 연결에 오류가 발생한다.<br>3b1. 네트워크 연결에 실패했다는 메시지를 출력한다. |

### Related Information

|             |            |
|-------------|------------|
| Performance | < 3 second |
|-------------|------------|

|             |      |
|-------------|------|
| Frequency   | None |
| Concurrency | None |
| Due Date    | None |

| Use Case #5 : User_image |  |
|--------------------------|--|
| General Characteristics  |  |
| Summary                  | MyPage를 이용하는 사용자들 개인만의 프로필 이미지를 추가/수정할 수 있다.                           |
| Scope                    |  |
| Level                    | User Level   |
| Author                   |  |
| Last Update              |  |
| Status                   | Analysis   |
| Primary Actor            | User   |
| Preconditions            | 사용자는 로그인이 되어 있어야 한다.<br>네트워크에 연결되어 있어야 한다.                             |
| Trigger                  | 사용자가 마이페이지로 이동하는 아이콘을 클릭한다   |
| Success Post Condition   | 사용자가 입력한 정보가 Server에 저장되어 있는 회원정보랑 일치하여 로그인에 성공한 상태여야 한다.              |
| Failed Post Condition    | 사용자가 입력한 정보가 Server에 저장되어 있는 회원정보랑 일치하지 않아 로그인에 성공한 상태가 아니라면 이용할 수 없다. |
| Main Success Scenario    |  |
| Step                     | Action   |
| 1                        | 이 Use case는 User가 마이페이지에서 프로필 이미지 수정을 시도한다.                            |
| 2                        | 회원 가입때 기입했던 ID와 Password를 토대로 로그인을 한 후, 마이페이지로 이동하는 아이콘을 클릭한다.         |
| 3                        | 프로필 이미지를 클릭한다.   |
| 4                        | 이미지를 클릭하면, 이미지를 추가하고 수정하는 탭을 사용할 수 있다.                                 |

| Extension Scenario  |  |
|---------------------|--|
| Step                | Branching Action   |
| 3                   | 3a. 사용자가 로그인이 되어있지 않다면 마이페이지 이동에 실패한다.<br>3a1. 마이페이지 이동 실패 메시지를 띄운다.<br>3a2. 로그인 페이지로 넘어가게 된다.<br>3b. 네트워크 연결에 오류가 발생한다.<br>3b1. 네트워크 연결에 실패했다는 메시지를 출력한다. |
| Related Information |  |
| Performance         | < 3 second   |
| Frequency           | None   |
| Concurrency         | None   |
| Due Date            | None   |

| Use Case #6 : User_nickname |  |
|-----------------------------|--|
| General Characteristics     |  |
| Summary                     | MyPage를 이용하는 사용자들 개인만의 닉네임을 수정할 수 있다.  |
| Scope                       |  |
| Level                       | User Level   |
| Author                      |  |
| Last Update                 |  |
| Status                      | Analysis   |
| Primary Actor               | User   |
| Preconditions               | 사용자는 로그인이 되어 있어야 한다.<br>네트워크에 연결되어 있어야 한다.   |
| Trigger                     | 사용자가 마이페이지로 이동하는 아이콘을 클릭한다   |
| Success Post Condition      | 사용자가 입력한 정보가 Server에 저장되어 있는 회원정보랑 일치하여 로그인에 성공한 상태여야 한다.  |
| Failed Post Condition       | 사용자가 입력한 정보가 Server에 저장되어 있는 회원정보랑 일치하지 않아 로그인에 성공한 상태가 아니라면 이용할 수 없다.   |
| Main Success Scenario       |  |
| Step                        | Action   |
| 1                           | 01 Use case는 User가 마이페이지에서 닉네임을 변경하기 위해 시도한다.  |
| 2                           | 회원 가입때 기입 했던 ID와 Password를 토대로 로그인을 한 후, 마이페이지로 이동하는 아이콘을 클릭한다.  |
| 3                           | 닉네임 배너를 클릭하면 닉네임을 변경할 수 있는 탭을 사용할 수 있다.  |
| Extension Scenario          |  |
| Step                        | Branching Action   |
| 3                           | 3a. 사용자가 로그인이 되어있지 않다면 마이페이지 이동에 실패한다.<br>3a1. 마이페이지 이동 실패 메시지를 띄운다.<br>3a2. 로그인 페이지로 넘어가게 된다.<br>3b. 네트워크 연결에 오류가 발생한다.<br>3b1. 네트워크 연결에 실패했다는 메시지를 출력한다. |
| Related Information         |  |
| Performance                 | < 3 second   |

|             |      |
|-------------|------|
| Frequency   | None |
| Concurrency | None |
| Due Date    | None |

| Use Case #7 : setPwd           |  |
|--------------------------------|--|
| <b>General Characteristics</b> |  |
| Summary                        | Mypage를 이용하는 사용자들 개인만의 비밀번호를 수정할 수 있다                                  |
| Scope                          |  |
| Level                          | User Level   |
| Author                         |  |
| Last Update                    |  |
| Status                         | Analysis   |
| Primary Actor                  | User   |
| Preconditions                  | 사용자는 로그인이 되어 있어야 한다.<br>네트워크에 연결되어 있어야 한다.                             |
| Trigger                        | 사용자가 마이페이지로 이동하는 아이콘을 클릭한다   |
| Success Post Condition         | 사용자가 입력한 정보가 Server에 저장되어 있는 회원정보랑 일치하여 로그인에 성공한 상태여야 한다.              |
| Failed Post Condition          | 사용자가 입력한 정보가 Server에 저장되어 있는 회원정보랑 일치하지 않아 로그인에 성공한 상태가 아니라면 이용할 수 없다. |
| <b>Main Success Scenario</b>   |  |
| Step                           | Action   |
| 1                              | 이 Use case는 User가 마이페이지에서 비밀번호를 변경하기 위해 시도한다.                          |
| 2                              | 회원 가입때 기입 했던 ID와 Password를 토대로 로그인을 한 후, 마이페이지로 이동하는 아이콘을 클릭한다.        |
| 3                              | 비밀번호 변경 배너를 클릭하면, 비밀번호를 변경할 수 있는 탭을 사용할 수 있다.                          |
| <b>Extension Scenario</b>      |  |
| Step                           | Branching Action   |
| 3                              | 3a. 사용자가 로그인이 되어있지 않다면, Use case 이용에 실패하게                              |

|  |  |
|--|--|
|  | <p>된다.</p> <p>3a1. 마이페이지 이동 실패 메시지를 띠운다.</p> <p>3a2. 로그인 페이지로 넘어가게 된다(Use case #1-2)</p> <p>3b. 비밀번호를 변경하는 과정에는 기존 비밀번호를 확인하는 과정이 있는데, 기존 비밀번호와 일치하지 않는 비밀번호를 기입한다면 비밀번호 수정에 실패한다.</p> <p>3b1. 기존 비밀번호와 일치하지 않다는 메시지를 띠운다.</p> <p>3c. 비밀번호를 변경하는 과정에서 변경하려는 비밀번호가 기존 비밀번호와 일치하다면 비밀번호 수정에 실패하게 된다.</p> <p>3c1. 기존 비밀번호와 다른 비밀번호를 기입하라는 메시지를 띠운다.</p> <p>3d. 네트워크 연결에 오류가 발생한다.</p> <p>3d1. 네트워크 연결에 실패했다는 메시지를 출력한다.</p> |
|--|--|

#### Related Information

|             |            |
|-------------|------------|
| Performance | < 3 second |
| Frequency   | None       |
| Concurrency | None       |
| Due Date    | None       |

#### Use Case #8 : myHistory

##### General Characteristics

|                        |  |
|------------------------|--|
| Summary                | MyPage를 이용하는 사용자들 개인만의 활동내역을 열람할 수 있다.                                 |
| Scope                  |  |
| Level                  | User Level   |
| Author                 |  |
| Last Update            |  |
| Status                 | Analysis   |
| Primary Actor          | User   |
| Preconditions          | 사용자는 로그인이 되어 있어야 한다.<br>네트워크에 연결되어 있어야 한다.                             |
| Trigger                | 사용자가 마이페이지로 이동하는 아이콘을 클릭한다   |
| Success Post Condition | 사용자가 입력한 정보가 Server에 저장되어 있는 회원정보랑 일치하여 로그인에 성공한 상태여야 한다.              |
| Failed Post Condition  | 사용자가 입력한 정보가 Server에 저장되어 있는 회원정보랑 일치하지 않아 로그인에 성공한 상태가 아니라면 이용할 수 없다. |
| Main Success Scenario  |  |

| Step                | Action   |
|---------------------|--|
| 1                   | 이 Use case는 User가 마이페이지에서 활동 내역을 열람하기 위해 시도한다.   |
| 2                   | 회원 가입때 기입 했던 ID와 Password를 토대로 로그인을 한 후, 마이페이지로 이동하게 해주는 아이콘을 클릭한다.  |
| 3                   | 활동내역 배너를 클릭하면 활동내역을 열람할 수 있는 탭을 사용할 수 있다.  |
| Extension Scenario  |  |
| Step                | Branching Action   |
| 3                   | 3a. 사용자가 로그인이 되어있지 않다면 마이페이지 이동에 실패한다.<br>3a1. 마이페이지 이동 실패 메시지를 띄운다.<br>3a2. 로그인 페이지로 넘어가게 된다.<br>3b. 네트워크 연결에 오류가 발생한다.<br>3b1. 네트워크 연결에 실패했다는 메시지를 출력한다. |
| Related Information |  |
| Performance         | < 3 second   |
| Frequency           | None   |
| Concurrency         | None   |
| Due Date            | None   |

### Use Case #9 : purchaseHistory

#### General Characteristics

|                        |  |
|------------------------|--|
| Summary                | MyPage를 이용하는 사용자들 개인만의 구매/대여 내역을 열람할 수 있다.                             |
| Scope                  |  |
| Level                  | User Level   |
| Author                 |  |
| Last Update            |  |
| Status                 | Analysis   |
| Primary Actor          | User   |
| Preconditions          | 사용자는 로그인이 되어 있어야 한다.<br>네트워크에 연결되어 있어야 한다.                             |
| Trigger                | 사용자가 마이페이지로 이동하는 아이콘을 클릭한다   |
| Success Post Condition | 사용자가 입력한 정보가 Server에 저장되어 있는 회원정보랑 일치하여 로그인에 성공한 상태여야 한다.              |
| Failed Post Condition  | 사용자가 입력한 정보가 Server에 저장되어 있는 회원정보랑 일치하지 않아 로그인에 성공한 상태가 아니라면 이용할 수 없다. |

#### Main Success Scenario

| Step | Action   |
|------|--|
| 1    | 이 Use case는 User가 마이페이지에서 구매/대여 내역을 열람하기 위해 시도한다.                  |
| 2    | 회원 가입때 기입 했던 ID와 Password를 토대로 로그인을 한 후, 마이페이지로 이동하게해주는 아이콘을 클릭한다. |
| 3    | 구매/대여 내역 배너를 클릭하면 활동내역을 열람할 수 있는 탭을 사용할 수 있다.                      |

#### Extension Scenario

| Step | Branching Action  |
|------|---|
| 3    | 3a. 네트워크에 문제가 있거나, 사용자가 로그인이 되어있지 않다면 마이페이지 이동에 실패한다.<br>3a1. 마이페이지 이동 실패 메시지를 띄운다.<br>3a2. 로그인 페이지로 넘어가게 된다.<br>3b1. 네트워크 연결에 실패했다는 메시지를 출력한다. |

#### Related Information

|             |            |
|-------------|------------|
| Performance | < 3 second |
|-------------|------------|

|             |      |
|-------------|------|
| Frequency   | None |
| Concurrency | None |
| Due Date    | None |

| Use Case #10 : User_group |  |
|---------------------------|--|
| General Characteristics   |  |
| Summary                   | MyPage를 이용하는 사용자들 개인의 “나의 그룹”을 열람할 수 있다.                               |
| Scope                     |  |
| Level                     | User Level   |
| Author                    |  |
| Last Update               |  |
| Status                    | Analysis   |
| Primary Actor             | User   |
| Preconditions             | 사용자는 로그인인이 되어 있어야 한다.<br>네트워크에 연결되어 있어야 한다.                            |
| Trigger                   | 사용자가 마이페이지로 이동하는 아이콘을 클릭한다   |
| Success Post Condition    | 사용자가 입력한 정보가 Server에 저장되어 있는 회원정보랑 일치하여 로그인에 성공한 상태여야 한다.              |
| Failed Post Condition     | 사용자가 입력한 정보가 Server에 저장되어 있는 회원정보랑 일치하지 않아 로그인에 성공한 상태가 아니라면 이용할 수 없다. |
| Main Success Scenario     |  |
| Step                      | Action   |
| 1                         | 이 Use case는 User가 마이페이지에서 나의 그룹을 열람하기 위해 시도한다.                         |
| 2                         | 회원 가입때 기입 했던 ID와 Password를 토대로 로그인을 한 후, 마이페이지로 이동하는 아이콘을 클릭한다.        |
| 3                         | 나의 그룹 배너를 클릭하면 나의 그룹을 열람할 수 있는 탭을 사용할 수 있다.                            |
| Extension Scenario        |  |
| Step                      | Branching Action   |
| 3                         | 3a. 사용자가 로그인이 되어있지 않다면 마이페이지 이동에 실패한다.                                 |

|                            |  |
|----------------------------|--|
|                            | 3a1. 마이페이지 이동 실패 메시지를 띠운다.<br>3a2. 로그인 페이지로 넘어가게 된다.<br>3b. 네트워크 연결에 오류가 발생한다.<br>3b1. 네트워크 연결에 실패했다는 메시지를 출력한다. |
| <b>Related Information</b> |  |
| Performance                | < 3 second   |
| Frequency                  | None   |
| Concurrency                | None   |
| Due Date                   | None   |

|                                |  |
|--------------------------------|--|
| Use Case #11 : Support         |  |
| <b>General Characteristics</b> |  |
| Summary                        | MyPage를 이용하는 사용자들 개인만의 문의 사항을 등록하거나 답변 완료 내역을 연락할 수 있다.                |
| Scope                          |  |
| Level                          | User Level   |
| Author                         |  |
| Last Update                    |  |
| Status                         | Analysis   |
| Primary Actor                  | User   |
| Preconditions                  | 네트워크에 연결되어 있어야 한다.<br>글을 등록하고자 하는 사용자는 회원가입이 되어 있어야 한다.                |
| Trigger                        | 사용자가 마이페이지로 이동하는 아이콘을 클릭한다   |
| Success Post Condition         | 사용자가 입력한 정보가 Server에 저장되어 있는 회원정보랑 일치하여 로그인에 성공한 상태여야 한다.              |
| Failed Post Condition          | 사용자가 입력한 정보가 Server에 저장되어 있는 회원정보랑 일치하지 않아 로그인에 성공한 상태가 아니라면 이용할 수 없다. |
| <b>Main Success Scenario</b>   |  |
| Step                           | Action   |
| 1                              | 0  Use case는 User가 Q&A기능을 사용하기 위해 시도한다.                                |
| 2                              | 회원 가입때 기입 했던 ID와 Password를 토대로 로그인을 한다.                                |

|   |   |
|---|---|
| 3 | 메인페이지에서 상단 메뉴의 Q&A를 클릭하면, Q&A를 등록하고 열람할 수 있는 페이지로 이동하게 된다.                |
| 4 | Q&A페이지에서는 글을 등록하거나 열람할 수 있다.<br>4.1 글을 등록하기 버튼을 클릭한다.<br>4.2 등록된 글을 클릭한다. |

#### Extension Scenario

| Step | Branching Action  |
|------|---|
| 3    | 3a. 사용자가 로그인이 되어있지 않다면 4.1의 글 등록하기에 실패한다.<br>3a1. 로그인 페이지로 넘어가게 된다.<br>3b. 네트워크 연결에 오류가 발생한다.<br>3b1. 네트워크 연결에 실패했다는 메시지를 출력한다. |

#### Related Information

|             |            |
|-------------|------------|
| Performance | < 3 second |
| Frequency   | None       |
| Concurrency | None       |
| Due Date    | None       |

#### Use Case #12 : Home

##### General Characteristics

|                        |                                     |
|------------------------|-------------------------------------|
| Summary                | MyPage를 이용하는 사용자들이 메인페이지로 이동할 수 있다. |
| Scope                  |                                     |
| Level                  | User Level                          |
| Author                 |                                     |
| Last Update            |                                     |
| Status                 | Analysis                            |
| Primary Actor          | User                                |
| Preconditions          | 네트워크에 연결되어 있어야 한다.                  |
| Trigger                | 사용자가 메인페이지로 이동하는 아이콘을 클릭한다          |
| Success Post Condition |                                     |
| Failed Post            |                                     |

|                              |  |
|------------------------------|--|
| Condition                    |  |
| <b>Main Success Scenario</b> |  |
| Step                         | Action   |
| 1                            | 01 Use case는 User가 메인페이지로 이동하기 위해 시도한다.                        |
| 2                            | 어떠한 페이지에서든 홈 이동 이미지를 클릭하면 페이지로 이동한다.                           |
| <b>Extension Scenario</b>    |  |
| Step                         | Branching Action   |
| 3                            | 3a. 네트워크에 문제가 있다면 이용에 실패한다..<br>3a1. 네트워크 연결에 실패했다는 메시지를 출력한다. |
| <b>Related Information</b>   |  |
| Performance                  | < 3 second   |
| Frequency                    | None   |
| Concurrency                  | None   |
| Due Date                     | None   |

|                                |  |
|--------------------------------|--|
| Use Case #13 : Log_out         |  |
| <b>General Characteristics</b> |  |
| Summary                        | 로그인된 사용자들이 로그아웃할 수 있다.   |
| Scope                          |  |
| Level                          | User Level   |
| Author                         |  |
| Last Update                    |  |
| Status                         | Analysis   |
| Primary Actor                  | User   |
| Preconditions                  | 사용자는 로그인이 되어 있어야 한다<br>네트워크에 연결되어 있어야 한다.  |
| Trigger                        | 사용자가 로그아웃 배너를 클릭한다.  |
| Success Post Condition         | 사용자가 입력한 정보가 Server에 저장되어 있는 회원정보랑 일치하여 로그인에 성공한 상태여야 한다.  |
| Failed Post Condition          |  |
| <b>Main Success Scenario</b>   |  |
| Step                           | Action   |
| 1                              | 0이 Use case는 User가 메인페이지로 이동하기 위해 시도한다.  |
| 2                              | 어떠한 페이지에서든 홈 이동 이미지를 클릭하면 메인페이지로 이동한다..  |
| <b>Extension Scenario</b>      |  |
| Step                           | Branching Action   |
| 3                              | 3a. 네트워크에 문제가 있다면 Use case 이용에 실패하게 된다.<br>3a1. 로그인 페이지로 넘어가게 된다.<br>3b. 로그인이 되어있지 않다면 Use case 이용에 실패하게 된다.<br>3b1. 아무런 반응이 없다. |
| <b>Related Information</b>     |  |
| Performance                    | < 3 second   |
| Frequency                      | None   |
| Concurrency                    | None   |

|          |      |
|----------|------|
| Due Date | None |
|----------|------|

| Use Case #14 : Classification  |   |
|--------------------------------|---|
| <b>General Characteristics</b> |   |
| Summary                        | 이 홈페이지를 이용하는 사용자들은 분야별, 테마별, 인기별로 분류된 도서 목록들을 통해 도서들을 추천받을 수 있다.          |
| Scope                          |   |
| Level                          | User Level  |
| Author                         |   |
| Last Update                    |   |
| Status                         | Analysis  |
| Primary Actor                  | User  |
| Preconditions                  | 네트워크에 연결되어 있어야 한다.  |
| Trigger                        | 사용자가 메인페이지에서 “추천 도서” 메뉴를 클릭한다   |
| Success Post Condition         |   |
| Failed Post Condition          |   |
| <b>Main Success Scenario</b>   |   |
| Step                           | Action  |
| 1                              | 01 Use case는 User가 도서 추천을 받기 위해 접근한다.                                     |
| 2                              | 메인페이지로 이동한다.  |
| 3                              | 메인페이지에서 “추천 도서”메뉴를 클릭하면, 추천 도서 페이지로 이동한다.                                 |
| 4                              | 기본적으로 임의의 도서들이 페이지에 나타나고, 분야별, 테마별, 인기별로 분류된 서브 메뉴를 통해 분류된 도서들을 확인할 수 있다. |
| <b>Extension Scenario</b>      |   |
| Step                           | Branching Action  |
| 3                              | 3a. 네트워크에 문제가 있다면, 페이지 이동에 실패한다.<br>3a1. 네트워크 연결에 실패했다는 메시지를 출력한다.        |
| <b>Related Information</b>     |   |
| Performance                    | < 3 second  |

|             |      |
|-------------|------|
| Frequency   | None |
| Concurrency | None |
| Due Date    | None |

| Use Case #15 : BookInfo |  |
|-------------------------|--|
| General Characteristics |  |
| Summary                 | 책의 상세 정보를 확인할 수 있는 페이지로 연동된다.                                      |
| Scope                   |  |
| Level                   | User Level   |
| Author                  |  |
| Last Update             |  |
| Status                  | Analysis   |
| Primary Actor           | User   |
| Preconditions           | 네트워크에 연결되어 있어야 한다.   |
| Trigger                 | “추천 도서”페이지에서 도서들을 클릭하거나 검색을 통해 접근한다.                               |
| Success Post Condition  |  |
| Failed Post Condition   |  |
| Main Success Scenario   |  |
| Step                    | Action   |
| 1                       | 0) Use case는 User가 책에 대한 상세 정보를 확인한다.                              |
| 2                       | 2.1 책 제목을 검색한다.<br>2.2 “추천 도서” 메뉴를 클릭한다.                           |
| 3                       | 3.1 검색된 책을 클릭한다.<br>3.2 추천 도서에 나타난 책을 클릭한다.                        |
| 4                       | 사용자는 책 표지, 책 소개, 목차 등 여러가지 상세정보가 나타나는 페이지로 이동된다.                   |
| Extension Scenario      |  |
| Step                    | Branching Action   |
| 3                       | 3a. 네트워크에 문제가 있다면 페이지 이동에 실패한다..<br>3a1. 네트워크 연결에 실패했다는 메시지를 출력한다. |

| Related Information |            |
|---------------------|------------|
| Performance         | < 3 second |
| Frequency           | None       |
| Concurrency         | None       |
| Due Date            | None       |

| Use Case #16 : Detail_Bookinfo |  |
|--------------------------------|--|
| General Characteristics        |  |
| Summary                        | 책을 대여하거나 구매를 희망할 때 보여지는 페이지이다.   |
| Scope                          |  |
| Level                          | User Level   |
| Author                         |  |
| Last Update                    |  |
| Status                         | Analysis   |
| Primary Actor                  | User   |
| Preconditions                  | 네트워크에 연결되어 있어야 한다.<br>사용자는 로그인이 되어 있어야 한다.                                 |
| Trigger                        | 책의 상세페이지에서 대여 또는 소장 버튼을 클릭한다.  |
| Success Post Condition         | 사용자가 입력한 정보가 Server에 저장되어 있는 회원정보랑 일치하여 로그인에 성공한 상태여야 한다.                  |
| Failed Post Condition          | 사용자가 입력한 정보가 Server에 저장되어 있는 회원정보와랑 일치하지 않아 로그인에 성공한 상태가 아니라면 이용할 수 없다.    |
| Main Success Scenario          |  |
| Step                           | Action   |
| 1                              | 0이 Use case는 User가 책을 대여하거나 구매하기 위해 사용한다.                                  |
| 2                              | 사용자는 로그인된 상태에서, “추천 도서” 메뉴를 통해 책의 상세 정보에 접근하거나, 검색을 통한 책의 상세 정보 페이지에 접근한다. |
| 3                              | 상세 정보 페이지에서 대여 또는 소장 버튼을 클릭한다.   |
| 4                              | 4.1 대여를 선택했을 경우 대여 일수를 선택하는 배너가 등장한다.<br>4.2 소장을 선택하면 구매페이지로 바로 넘어가게 된다.   |
| Extension Scenario             |  |

| Step                | Branching Action  |
|---------------------|---|
| 3                   | 3a. 사용자가 로그인이 되어있지 않다면 Use case 이용에 실패하게 된다.<br>3a1. 로그인 오류 메시지를 띄운다.<br>3b. 네트워크 연결에 오류가 발생한다.<br>3b1. 네트워크 연결에 실패했다는 메시지를 출력한다. |
| Related Information |   |
| Performance         | < 3 second  |
| Frequency           | None  |
| Concurrency         | None  |
| Due Date            | None  |

| Use Case #17 : Subscribe |   |
|--------------------------|---|
| General Characteristics  |   |
| Summary                  | MyPage를 이용하는 사용자들을 위한 정기대여권 구매 페이지이다.                                   |
| Scope                    |   |
| Level                    | User Level  |
| Author                   |   |
| Last Update              |   |
| Status                   | Analysis  |
| Primary Actor            | User  |
| Preconditions            | 네트워크에 연결되어 있어야 한다.<br>사용자는 로그인이 되어 있어야 한다.                              |
| Trigger                  | 사용자가 멤버십 가입 버튼을 클릭한다.   |
| Success Post Condition   | 사용자가 입력한 정보가 Server에 저장되어 있는 회원정보랑 일치하여 로그인에 성공한 상태여야 한다.               |
| Failed Post Condition    | 사용자가 입력한 정보가 Server에 저장되어 있는 회원정보와랑 일치하지 않아 로그인에 성공한 상태가 아니라면 이용할 수 없다. |
| Main Success Scenario    |   |
| Step                     | Action  |

|                            |   |
|----------------------------|---|
| 1                          | 01 Use case는 User가 멤버십 가입을 위해 시도한다.   |
| 2                          | 2.1 메인 메뉴에서 정기권 구매 버튼을 한다.<br>2.2 책을 대여, 구매하는 과정에서 정기권 구매 버튼을 클릭한다.<br>2.3 장바구니 버튼을 클릭하면 장바구니에 담기게 된다,                            |
| 3                          | 버튼을 클릭하면 정기권을 구매하는 페이지로 넘어가게 된다.  |
| <b>Extension Scenario</b>  |   |
| Step                       | Branching Action  |
| 3                          | 3a. 사용자가 로그인이 되어있지 않다면 Use case 이용에 실패하게 된다.<br>3a1. 로그인 오류 메시지를 띠운다.<br>3b. 네트워크 연결에 오류가 발생한다.<br>3b1. 네트워크 연결에 실패했다는 메시지를 출력한다. |
| <b>Related Information</b> |   |
| Performance                | < 3 second  |
| Frequency                  | None  |
| Concurrency                | None  |
| Due Date                   | None  |

|                                |  |
|--------------------------------|--|
| Use Case #18 : Shopping_cart   |  |
| <b>General Characteristics</b> |  |
| Summary                        | 책을 대여하거나 구매하기 전에 장바구니에 담아 보관할 수 있다.        |
| Scope                          |  |
| Level                          | User Level                                 |
| Author                         |  |
| Last Update                    |  |
| Status                         | Analysis                                   |
| Primary Actor                  | User                                       |
| Preconditions                  | 네트워크에 연결되어 있어야 한다.<br>사용자는 로그인이 되어 있어야 한다. |
| Trigger                        | 사용자가 장바구니 메뉴를 클릭한다.                        |

|                               |   |
|-------------------------------|---|
| <b>Success Post Condition</b> | 사용자가 입력한 정보가 Server에 저장되어 있는 회원정보랑 일치하여 로그인에 성공한 상태여야 한다.   |
| <b>Failed Post Condition</b>  | 사용자가 입력한 정보가 Server에 저장되어 있는 회원정보와랑 일치하지 않아 로그인에 성공한 상태가 아니라면 이용할 수 없다.   |
| <b>Main Success Scenario</b>  |   |
| <b>Step</b>                   | <b>Action</b>   |
| 1                             | 01 Use case는 User가 도서를 대여 또는 소장하기 전 보관을 한다.   |
| 2                             | User는 로그인 된 상태에서, 추천 도서 배너를 통한 책의 상세 정보에 접근하거나, 검색을 통한 책의 상세 정보 페이지에 접근한다.  |
| 3                             | 3.1 상세 정보에서 장바구니 담기 버튼을 클릭한다.<br>3.2 구매 페이지에서 장바구니 담기 버튼을 클릭한다.   |
| <b>Extension Scenario</b>     |   |
| <b>Step</b>                   | <b>Branching Action</b>   |
| 3                             | 3a. 사용자가 로그인이 되어있지 않다면 Use case 이용에 실패하게 된다.<br>3a1. 로그인 오류 메시지를 띄운다.<br>3b. 네트워크 연결에 오류가 발생한다.<br>3b1. 네트워크 연결에 실패했다는 메시지를 출력한다. |
| <b>Related Information</b>    |   |
| <b>Performance</b>            | < 3 second  |
| <b>Frequency</b>              | None  |
| <b>Concurrency</b>            | None  |
| <b>Due Date</b>               | None  |

|                                |                                   |
|--------------------------------|-----------------------------------|
| Use Case #19 : Purchase_page   |                                   |
| <b>General Characteristics</b> |                                   |
| <b>Summary</b>                 | 책을 대여하거나 소장하기 위한 결제를 하기 위한 페이지이다. |
| <b>Scope</b>                   |                                   |
| <b>Level</b>                   | User Level                        |
| <b>Author</b>                  |                                   |
| <b>Last Update</b>             |                                   |

|                        |   |
|------------------------|---|
| Status                 | Analysis  |
| Primary Actor          | User  |
| Preconditions          | 네트워크에 연결되어 있어야 한다.<br>글을 등록하고자 하는 사용자는 회원가입이 되어 있어야 한다.   |
| Trigger                | 사용자가 마이페이지에서 구매/대여 내역 배너를 클릭한다.   |
| Success Post Condition | 사용자가 입력한 정보가 Server에 저장되어 있는 회원정보랑 일치하여 로그인에 성공한 상태여야 한다.   |
| Failed Post Condition  | 사용자가 입력한 정보가 Server에 저장되어 있는 회원정보랑 일치하지 않아 로그인에 성공한 상태가 아니라면 이용할 수 없다.  |
| Main Success Scenario  |   |
| Step                   | Action  |
| 1                      | 01 Use case는 User가 마이페이지에서 구매/대여를 하기 위한 페이지이다.  |
| 2                      | User는 로그인 된 상태에서 추천 도서 배너를 통한 책의 상세 정보에 접근하거나, 검색을 통한 책의 상세 정보 페이지에 접근한다.   |
| 3                      | 대여 또는 소장버튼을 통해 #13 Detail_Bookinfo 과정을 거친다.   |
| 4                      | #13 Detail_Bookinfo 과정을 지나면 구매페이지가 등장한다.  |
| Extension Scenario     |   |
| Step                   | Branching Action  |
| 3                      | 3a. 사용자가 로그인이 되어있지 않다면 4.1의 글 등록하기에 실패한다.<br>3a1. 로그인 페이지로 넘어가게 된다.<br>3b. 네트워크 연결에 오류가 발생한다.<br>3b1. 네트워크 연결에 실패했다는 메시지를 출력한다. |
| Related Information    |   |
| Performance            | < 3 second  |
| Frequency              | None  |
| Concurrency            | None  |
| Due Date               | None  |

| Use Case #20 : FileViewer |   |
|---------------------------|---|
| General Characteristics   |   |
| Summary                   | 책을 열람할 수 있는 페이지이며, 사용자 메뉴를 통해 열람에 도움이 되는 기능들을 사용할 수 있다.   |
| Scope                     |   |
| Level                     | User Level  |
| Author                    |   |
| Last Update               |   |
| Status                    | Analysis  |
| Primary Actor             | User  |
| Preconditions             | 네트워크에 연결되어 있어야 한다.<br>클을 등록하고자 하는 사용자는 회원가입이 되어 있어야 한다.<br>사용자는 정기 대여권을 갖고 있거나,<br>대여 또는 소장을 한 상태여야 한다.                                     |
| Trigger                   | 사용자가 책 상세 정보 페이지에서 책 읽기 버튼을 클릭한다.   |
| Success Post Condition    | 사용자가 입력한 정보가 Server에 저장되어 있는 회원정보랑 일치하여<br>로그인에 성공한 상태여야 한다.  |
| Failed Post Condition     | 사용자가 입력한 정보가 Server에 저장되어 있는 회원정보랑<br>일치하지 않아<br>로그인에 성공한 상태가 아니라면 이용할 수 없다.<br>사용자는 정기 대여권을 갖고 있지 않거나,<br>대여 또는 소장을 하지 않았다면, 버튼이 노출되지 않는다. |
| Main Success Scenario     |   |
| Step                      | Action  |
| 1                         | 0) Use case는 정기권을 보유중이거나, 대여 또는 소장을 한 User들을 위해, 책을 열람하기 위한 페이지이다.  |
| 2                         | User는 로그인 된 상태에서 추천 도서 배너를 통한 책의 상세 정보에 접근하거나, 검색을 통한 책의 상세 정보 페이지에 접근한다.   |
| 3                         | 대여 또는 소장을 한 후, 노출된 책 읽기 버튼을 클릭하여 접근한다.  |
| Extension Scenario        |   |
| Step                      | Branching Action  |
| 3                         | 3a. 사용자가 로그인이 되어있지 않다면 로그인 실패 메시지를 띠운다.<br>3a1. 로그인 페이지로 넘어가게 된다.   |

|                            |   |
|----------------------------|---|
|                            | <p>3b. 네트워크 연결에 오류가 발생한다.<br/>         3b1. 네트워크 연결에 실패했다는 메시지를 출력한다.</p> <p>3c. 대여나 소장을 하지 않았거나, 정기권이 없다면 책 읽기 버튼이 노출되지 않는다.</p> |
| <b>Related Information</b> |   |
| Performance                | < 3 second  |
| Frequency                  | None  |
| Concurrency                | None  |
| Due Date                   | None  |

|                                |  |
|--------------------------------|--|
| Use Case #21 : GroupList       |  |
| <b>General Characteristics</b> |  |
| Summary                        | 이 홈페이지를 이용하는 사용자가 그룹 목록을 확인할 수 있도록 하는 페이지이다. |
| Scope                          |  |
| Level                          | User Level                                   |
| Author                         |  |
| Last Update                    |  |
| Status                         | Analysis                                     |
| Primary Actor                  | User   |
| Preconditions                  | 네트워크에 연결되어 있어야 한다.                           |
| Trigger                        | 사용자가 그룹 페이지로 이동하는 버튼을 누른다.                   |

|                              |  |
|------------------------------|--|
| Success Post Condition       |  |
| Failed Post Condition        |  |
| <b>Main Success Scenario</b> |  |
| Step                         | Action   |
| 1                            | 이 Use case는 사용자가 그룹 목록을 확인할 때 사용한다.              |
| 2                            | 메인 배너의 그룹 페이지로 이동하는 버튼을 누른다.                     |
| 3                            | 그룹 페이지로 이동하면 그룹 목록을 볼 수 있다.                      |
| <b>Extension Scenario</b>    |  |
| Step                         | Branching Action                                 |
| 3                            | 3a. 네트워크에 문제가 있으면 이동에 실패한다.<br>3a1. 에러 페이지를 띄운다. |
| <b>Related Information</b>   |  |
| Performance                  | < 3 second                                       |
| Frequency                    | None   |
| Concurrency                  | None   |
| Due Date                     | None   |

## Use Case #22 : GroupInfo

### General Characteristics

|                        |  |
|------------------------|--|
| Summary                | 이 홈페이지를 이용하는 사용자가 그룹 소개를 확인할 수 있는 페이지이다. |
| Scope                  |  |
| Level                  | User Level                               |
| Author                 |  |
| Last Update            |  |
| Status                 | Analysis                                 |
| Primary Actor          | User                                     |
| Preconditions          | 네트워크에 연결되어 있어야 한다.                       |
| Trigger                | 사용자가 그룹 목록에서 해당 그룹으로 이동하는 버튼을 누른다.       |
| Success Post Condition |  |
| Failed Post Condition  |  |
| Main Success Scenario  |  |
| Step                   | Action                                   |

|   |                                     |
|---|-------------------------------------|
| 1 | 이 Use case는 사용자가 그룹 설명을 확인할 때 사용한다. |
| 2 | 그룹 목록에서 해당 그룹을 클릭한다.                |
| 3 | 그룹 소개 페이지로 이동하면 그룹 소개를 볼 수 있다.      |

#### Extension Scenario

| Step | Branching Action                                |
|------|---|
| 3    | 3a. 네트워크에 문제가 있으면 이동에 실패한다.<br>3a1. 여러 팝업을 띄운다. |

#### Related Information

|             |            |
|-------------|------------|
| Performance | < 3 second |
| Frequency   | None       |
| Concurrency | None       |
| Due Date    | None       |

#### Use Case #23 : GroupCreation

#### General Characteristics

|         |                                 |
|---------|---------------------------------|
| Summary | 이 홈페이지를 이용하는 사용자가 그룹을 생성할 수 있다. |
| Scope   |                                 |

| Level                  | User Level                       |
|------------------------|----------------------------------|
| Author                 |                                  |
| Last Update            |                                  |
| Status                 | Analysis                         |
| Primary Actor          | Member                           |
| Preconditions          | 네트워크에 연결되어 있고, 로그인되어 있어야 한다.     |
| Trigger                | 사용자가 그룹을 생성하는 버튼을 누른다.           |
| Success Post Condition | 사용자는 그룹을 생성할 수 있다.               |
| Failed Post Condition  | 그룹을 생성할 수 없다.                    |
| Main Success Scenario  |                                  |
| Step                   | Action                           |
| 1                      | 이 Use case는 사용자가 그룹을 생성할 때 이용한다. |
| 2                      | 그룹 목록 페이지에서 그룹을 생성하는 버튼을 누른다.    |
| 3                      | 그룹 생성 페이지로 이동하면 그룹을 생성할 수 있다.    |
| 4                      | 사용자는 그룹 이름, 그룹 설명을 입력하고 제출한다.    |

|                     |   |
|---------------------|---|
| 5                   | 서버로 그룹 이름, 그룹 설명이 전송되고 세션에 저장된 사용자가 자동으로 리더가 되고 모집상태는 모집 중으로 초기화되어 저장된다.  |
| 6                   | 서버가 DB로 전송하면 그룹이 생성된다.  |
| Extension Scenario  |   |
| Step                | Branching Action  |
| 3                   | 3a. 네트워크에 문제가 있으면 이동에 실패한다.<br>3a1. 에러 팝업을 띄운다.<br>3b. 로그인이 되지 않은 사용자는 이동에 실패한다.<br>3b1. 에러 팝업을 띄운다.<br>3b2. 로그인 페이지로 이동한다.                   |
| 4                   | 4a. 입력 데이터가 부족한 경우 생성에 실패한다.<br>4a1. 에러 팝업을 띄운다.<br>4a2. 다시 입력할 수 있게 한다.<br>4b. 네트워크에 문제가 있으면 생성에 실패한다.<br>4b1. 에러 팝업을 띄운다.<br>4b2. 홈으로 돌아간다. |
| 6                   | 6a. 네트워크에 문제가 있으면 생성에 실패한다.<br>6a1. 에러 팝업을 띄운다.<br>6a2. 홈으로 이동한다.   |
| Related Information |   |
| Performance         | < 3 second  |

|             |      |
|-------------|------|
| Frequency   | None |
| Concurrency | None |
| Due Date    | None |

| Use Case #24: GroupManage |                                 |
|---------------------------|---------------------------------|
| General Characteristics   |                                 |
| Summary                   | 이 홈페이지를 이용하는 사용자는 그룹을 관리할 수 있다. |
| Scope                     |                                 |
| Level                     | User Level                      |
| Author                    |                                 |
| Last Update               |                                 |
| Status                    | Analysis                        |
| Primary Actor             | User                            |
| Preconditions             | 네트워크에 연결되어 있고, 로그인되어 있어야 한다.    |
| Trigger                   | 사용자가 그룹에 가입하는 버튼을 누른다.          |
| Success Post Condition    | 사용자는 선택한 그룹의 관리자여야              |

|                              | 한다.                                       |
|------------------------------|---|
| Failed Post Condition        | 사용자는 선택한 그룹의 관리자가 아니라면 이용할 수 없다.          |
| <b>Main Success Scenario</b> |   |
| Step                         | Action                                    |
| 1                            | 사용자가 그룹을 관리할 때 사용한다.                      |
| 2                            | 그룹 페이지에서 그룹 관리를 누른다.                      |
| 3                            | 그룹 관리 페이지에서 해당 그룹에 들어온 가입 신청 목록을 볼 수 있다.  |
| 4                            | 받아줄 유저를 선택하고 받아주기 버튼을 누른다.                |
| 5                            | 해당 유저는 그룹의 그룹원이 된다.                       |
| <b>Extension Scenario</b>    |   |
| Step                         | Branching Action                          |
| 2                            | 2a. 사용자가 해당 그룹의 관리자가 아닌 경우 버튼을 노출시키지 않는다. |

|   |   |
|---|---|
| 3 | <p>3a. 네트워크에 문제가 있으면 이동에 실패한다.</p> <p>3a1. 에러 팝업을 띄운다.</p> <p>3a2. 훔으로 이동한다.</p> <p>3b. 사용자가 로그인 되지 않았거나 현재 그룹의 관리자가 아니라면 이용할 수 없다.</p> <p>3b1. 에러 팝업을 띄운다.</p> <p>3b2. 이전 페이지로 이동한다.</p> |
|---|---|

#### Related Information

|             |            |
|-------------|------------|
| Performance | < 3 second |
| Frequency   | None       |
| Concurrency | None       |
| Due Date    | None       |

#### Use Case #25 : GroupPlan

#### General Characteristics

|         |   |
|---------|---|
| Summary | 이 홈페이지를 이용하는 사용자가 해당 그룹에 대해 플랜을 생성하는 페이지이다. |
| Scope   |   |
| Level   | User Level                                  |

| Author                 |                                     |
|------------------------|-------------------------------------|
| Last Update            |                                     |
| Status                 | Analysis                            |
| Primary Actor          | User                                |
| Preconditions          | 네트워크에 연결되어 있어야 한다.                  |
| Trigger                | 그룹 관리에서 플랜을 생성하는 버튼을 누른다.           |
| Success Post Condition | 사용자는 해당 그룹의 멤여야 한다.                 |
| Failed Post Condition  | 해당 그룹의 멤버가 아닌 경우 이용할 수 없다.          |
| Main Success Scenario  |                                     |
| Step                   | Action                              |
| 1                      | 이 Use case는 사용자가 그룹 플랜을 생성할 때 사용한다. |
| 2                      | 그룹 페이지에서 플랜 생성 버튼을 누른다.             |
| 3                      | 플랜 생성 페이지로 이동하면 플랜 작성 서식을 작성할 수 있다. |
| 4                      | 사용자가 플랜 제목을 입력하고 제출한다.              |
| 5                      | 서버는 해당 데이터를 받아 DB로 전송한다.            |
| Extension Scenario     |                                     |

| Step | Branching Action                                    |
|------|---|
| 2    | 2a. 그룹의 멤버가 아닌 경우 플랜 생성하기 버튼을 노출시키지 않는다.            |
| 3    | 3a. 네트워크에 문제가 있으면 이동에 실패한다.<br>3a1. 에러 페이지를 띄운다.    |
| 4    | 4a. 플랜 제목을 입력하지 않은 경우 생성에 실패한다.<br>4a1. 에러 팝업을 띄운다. |

#### Related Information

|             |            |
|-------------|------------|
| Performance | < 3 second |
| Frequency   | None       |
| Concurrency | None       |
| Due Date    | None       |

#### Use Case #26 : Group Join

##### General Characteristics

|               |                     |
|---------------|---------------------|
| Summary       | 사용자가 그룹에 가입하는 경우이다. |
| Scope         |                     |
| Level         | Member level        |
| Author        |                     |
| Last Update   |                     |
| Status        | Analysis            |
| Primary Actor | User                |

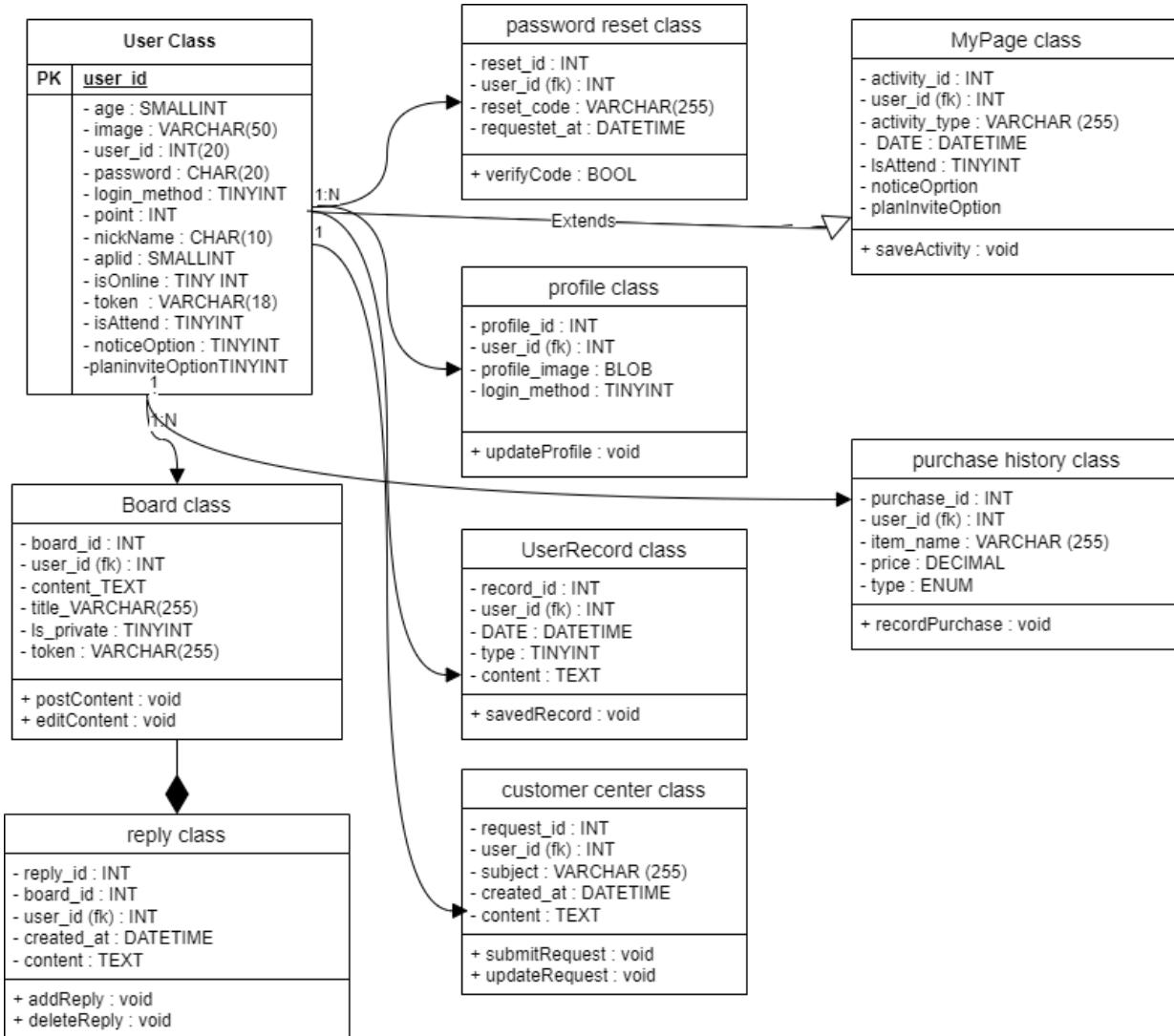
|                               |   |
|-------------------------------|---|
| <b>Preconditions</b>          | 사용자는 로그인이 되어 있어야 한다<br>네트워크에 연결되어 있어야 한다.<br>사용자는 그 그룹에 가입되어 있지 않아야 한다. |
| <b>Trigger</b>                | 사용자가 그룹 참여하기 버튼을 누른다.   |
| <b>Success Post Condition</b> | 사용자가 그룹 가입을 하거나 신청을 보낼 수 있다.  |
| <b>Failed Post Condition</b>  | 사용자가 그룹에 가입하지 못한다.  |
| <b>Main Success Scenario</b>  |   |
| <b>Step</b>                   | <b>Action</b>   |
| 1                             | 0이 usecase는 사용자가 그룹에 가입할 때 사용된다.  |
| 2                             | 그룹 상세 페이지에서 참여하기 버튼을 누른다.   |
| 3                             | 그룹 가입 조건이 자유인 경우 사용자는 그룹 멤버 테이블에 추가된다.                                  |
| 4                             | 그룹 가입 조건이 신청인 경우 사용자는 그룹 가입 신청 테이블에 추가된다.                               |
| <b>Extension Scenario</b>     |   |
| <b>Step</b>                   | <b>Branching Action</b>   |
| 2                             | 2a. 로그인 되지 않았거나 이미 그룹에 가입한 경우 참여하기 버튼을 노출시키지 않는다.                       |
| 3                             | 3a. 네트워크에 문제가 있다면 Use case 이용에 실패하게 된다.<br>3a1. 홈 페이지로 넘어가게 된다.         |
| <b>Related Information</b>    |   |
| <b>Performance</b>            | < 3 second  |
| <b>Frequency</b>              | None  |
| <b>Concurrency</b>            | None  |

|                                   |                       |
|-----------------------------------|-----------------------|
| <b>Use Case #27: Plan content</b> |                       |
| <b>General Characteristics</b>    |                       |
| <b>Summary</b>                    | 그룹 플랜에 독후감을 작성할 수 있다. |
| <b>Scope</b>                      |                       |
| <b>Level</b>                      | User Level            |

|                              |  |
|------------------------------|--|
| Author                       |  |
| Last Update                  |  |
| Status                       | Analysis   |
| Primary Actor                | User   |
| Preconditions                | 사용자는 로그인이 되어 있어야 한다.<br>네트워크에 연결되어 있어야 한다.<br>사용자는 해당 그룹의 멤버여야 한다.             |
| Trigger                      | 사용자가 플랜의 글쓰기 버튼을 누른다.  |
| Success Post Condition       | 플랜에 사용자의 독후감이 저장된다.  |
| Failed Post Condition        | 사용자는 독후감을 작성할 수 없다.  |
| <b>Main Success Scenario</b> |  |
| Step                         | Action   |
| 1                            | 이 use case는 사용자가 플랜에 독후감을 작성할 때 사용된다.  |
| 2                            | 사용자가 그룹 페이지에서 플랜을 선택하고 글쓰기 버튼을 누르면 시작된다.                                       |
| 3                            | 서버는 해당 그룹의 책에 대하여 사용자가 남긴 감상평을 모두 불러온다.  |
| 4                            | 사용자는 자신이 쓴 감상평을 선택하여 독후감에 추가할 수 있다.  |
| 5                            | 사용자는 제출하기 버튼을 누른다.   |
| 6                            | 서버에 해당 플랜의 감상평이 전송된다.  |
| <b>Extension Scenario</b>    |  |
| Step                         | Branching Action   |
| 3                            | 3a. 네트워크에 문제가 있다면 Use case 이용에 실패하게 된다.<br>3a1. 흘 페이지로 이동한다.                   |
| 5                            | 5a. 감상평을 작성하지 않은 경우 전송에 실패한다.<br>5a1. 에러 팝업을 띄운다.<br>5a2. 독후감을 다시 작성할 수 있게 한다. |
| <b>Related Information</b>   |  |
| Performance                  | < 3 second   |
| Frequency                    | None   |
| Concurrency                  | None   |

### 3. Class diagram

– Draw class diagrams.  
 Describe each class in detail (attributes, methods, others) (table type).  
 12pt, 160%.



위 다이어그램은 설계단계에서의 생각했던 테이블과 그에따른 역할들과 매개변수와 함수이다.

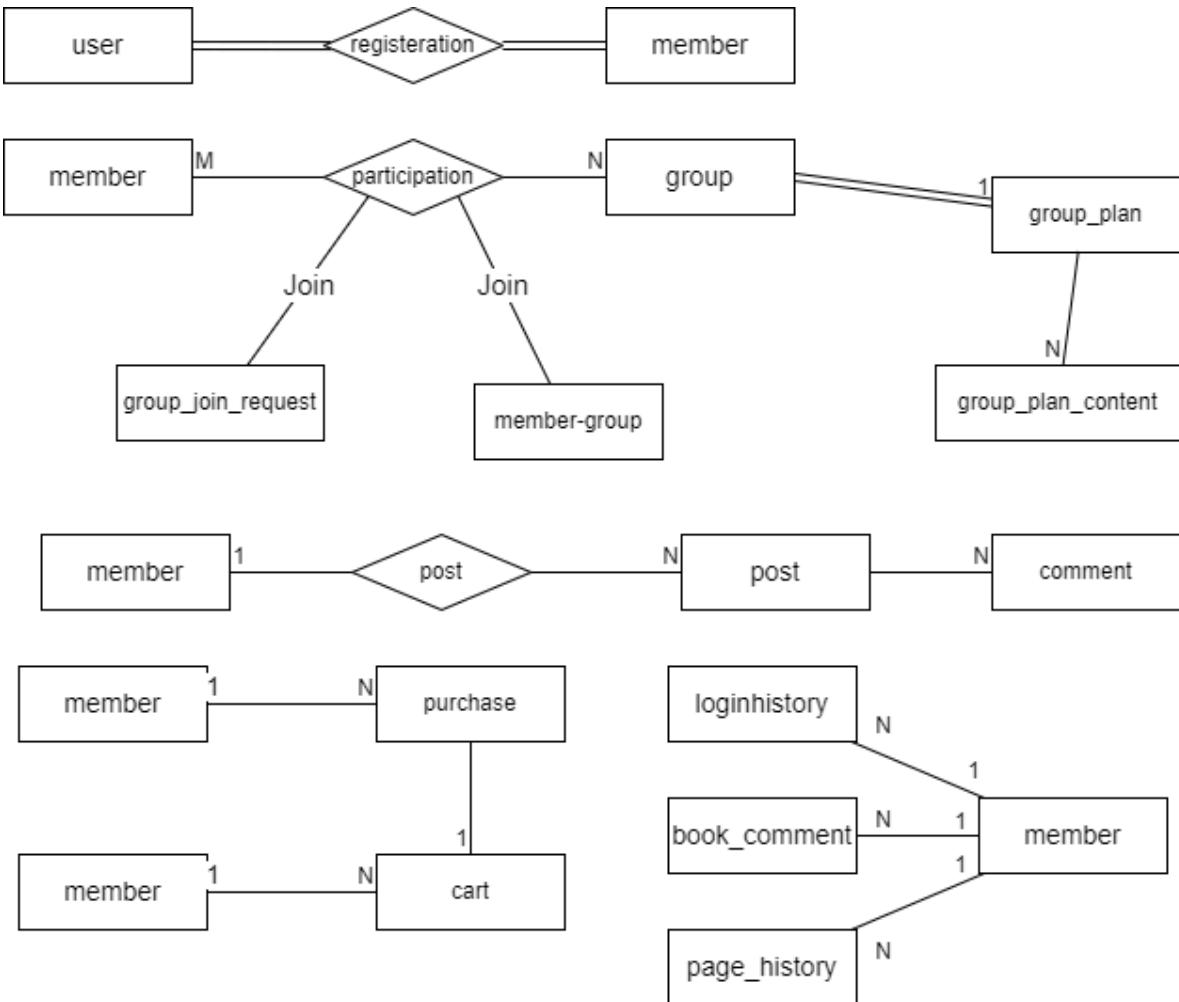
하지만, 최종 구현단계에선 설계단계때와는 조금 다른 방식으로 구현이 되었는데 실제로 구현해야하는 내용을 직접 ER다이어그램을 그려보며 이해하는 방식을 사용했다.

또한, 보고서 작성에도 ER다이어그램을 피터첸 표기법을 활용하여 개념적 모델로 정리하며 구현하는방식을 활용했다.

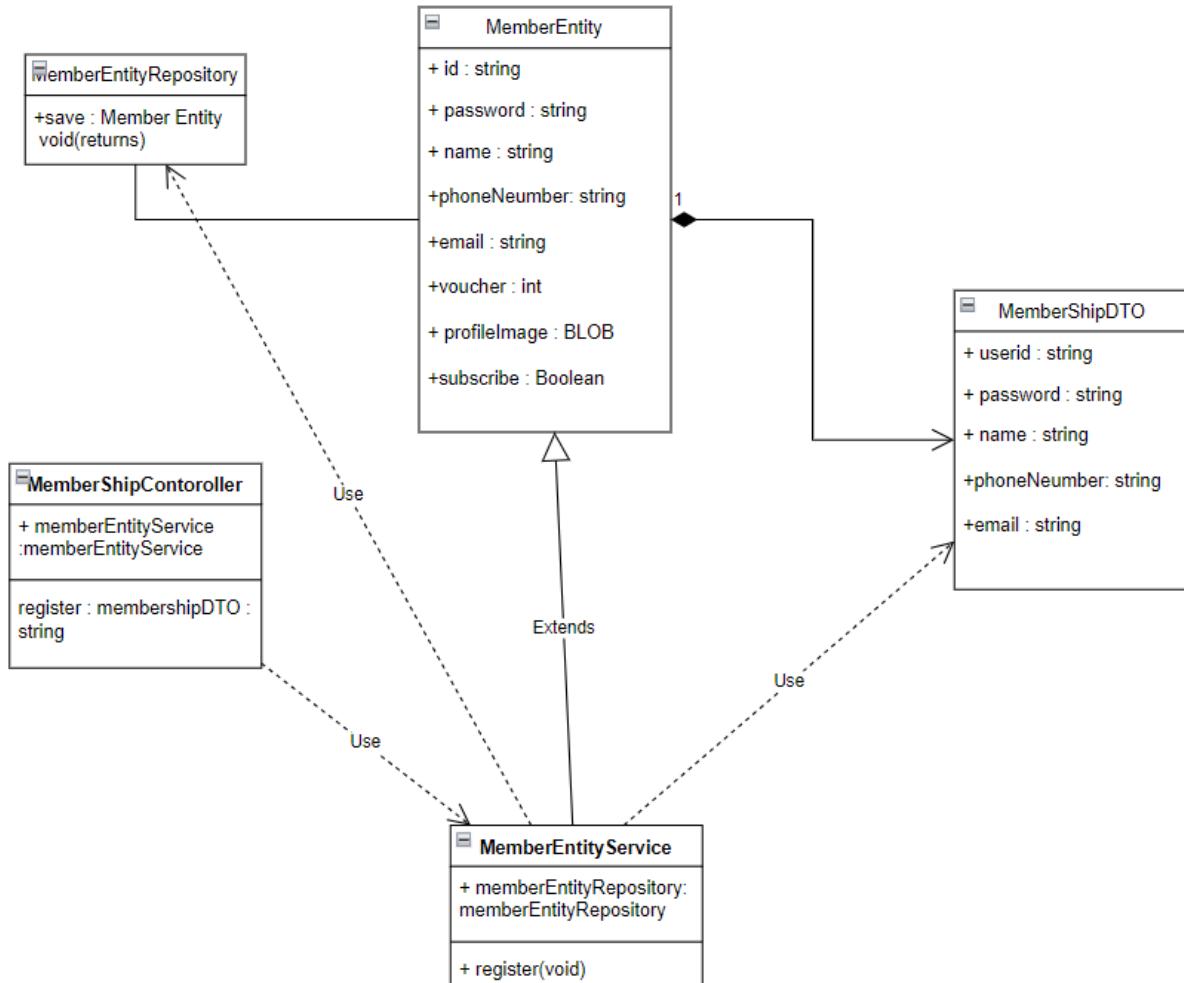
Key, 속성, 관계등을 정리한 정규화 과정을 세부적으로 알아보는 논리적인 데이터 모델링을 통해 테이블과 테이블의 종류 테이블의 키를 나타내고, 그 관계에 관련한 기능에 대해 디테일한 설명을 추가한것이다. DB다이어그램을 개념적 설계 모델로 작성해 보았고 디테일한 설계방식에 대한 내용은 클래스 다이어그램에 추가했다.

설계단계와는 다른 실제 구현된 유의미한 기능을 소개하겠다.

### <개념설 설계 ER diagram>



### 1. 회원가입 관련 클래스 다이어그램



## MemberEntity

|                   |                      |         |            |
|-------------------|----------------------|---------|------------|
| Class Description | 회원 정보를 저장하는 엔티티 클래스. |         |            |
| 구분                | Name                 | Type    | Visibility |
|                   | Description          |         |            |
| Attributes        | <b>id</b>            | string  | private    |
|                   | <b>password</b>      | string  | private    |
|                   | <b>name</b>          | string  | private    |
|                   | <b>phoneNumber</b>   | string  | private    |
|                   | <b>email</b>         | string  | private    |
|                   | <b>voucher</b>       | int     | private    |
|                   | <b>profileImage</b>  | string  | private    |
|                   | <b>subscribe</b>     | boolean | private    |

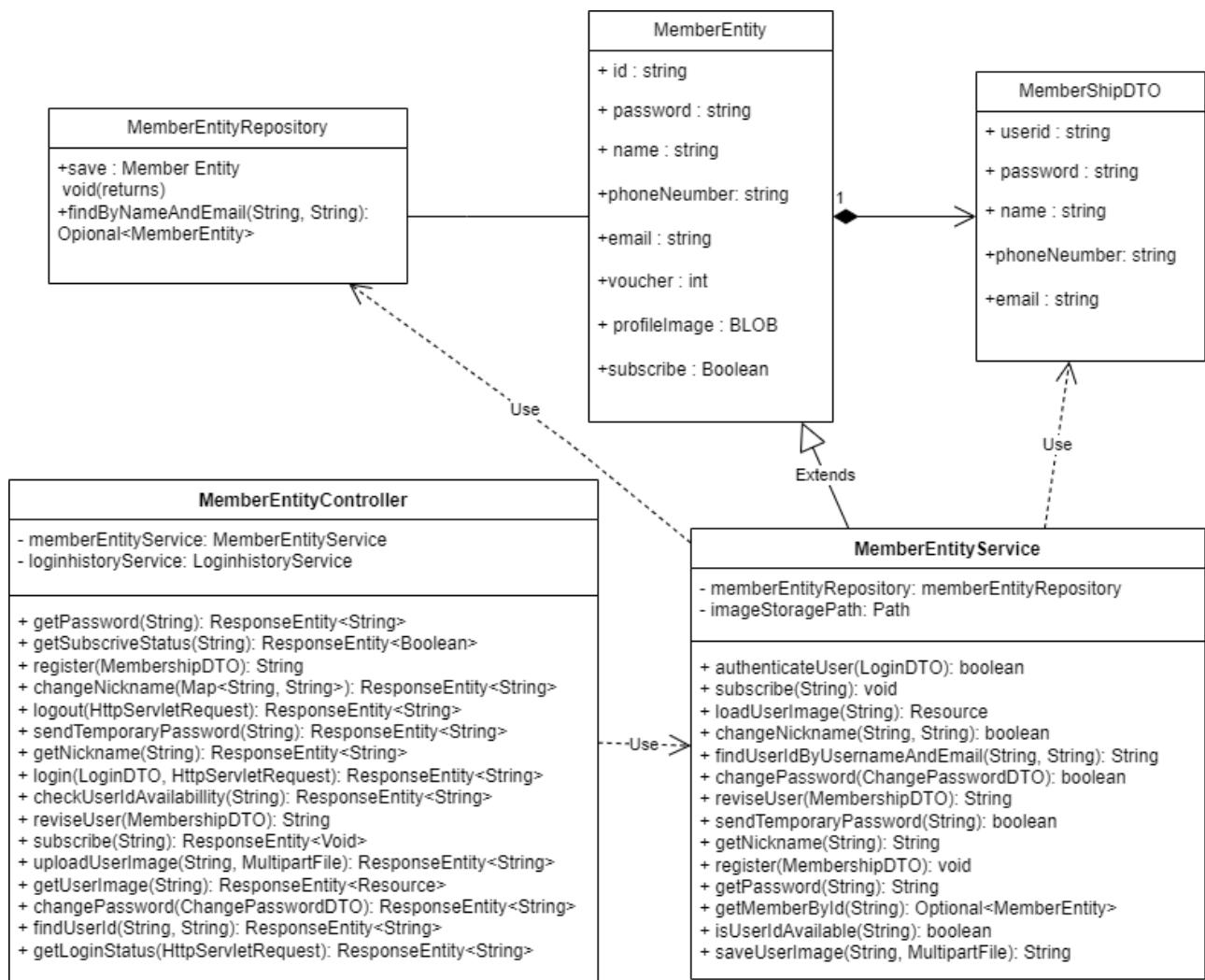
| <b>MembershipDTO</b> |                             |          |            |
|----------------------|-----------------------------|----------|------------|
| Class Description    | 회원가입 요청 시 데이터를 전달하기 위한 DTO. |          |            |
| 구분                   | Name                        | Type     | Visibility |
|                      | Description                 |          |            |
| Attributes           | <b>userid</b>               | string   | private    |
|                      | <b>password</b>             | string   | private    |
|                      | <b>name</b>                 | string   | private    |
|                      | <b>phoneNumber</b>          | string   | private    |
|                      | <b>email</b>                | string   | private    |
| 구분                   | Name                        | Argument | Returns    |
|                      | Description                 |          |            |
| Operations           |                             |          |            |

| <b>MemberShip Controller</b> |                                  |                               |            |
|------------------------------|----------------------------------|-------------------------------|------------|
| Class Description            | 회원가입과 관련된 비즈니스 로직을 처리하는 서비스 클래스. |                               |            |
| 구분                           | Name                             | Type                          | Visibility |
|                              | Description                      |                               |            |
| Attributes                   | <b>memberEntityRepository</b>    | <b>memberEntityRepository</b> | private    |
| 구분                           | Name                             | Argument                      | Returns    |
|                              | Description                      |                               |            |
| Operations                   | <b>register</b>                  | MembershipDTO                 | void       |

| <b>MemberEntityService</b> |                            |                            |            |
|----------------------------|----------------------------|----------------------------|------------|
| Class Description          | 회원가입 요청을 처리하는 컨트롤러 클래스.    |                            |            |
| 구분                         | Name                       | Type                       | Visibility |
|                            | Description                |                            |            |
| Attributes                 | <b>memberEntityService</b> | <b>memberEntityService</b> | private    |

| 구분         | Name            | Argument      | Returns |  |
|------------|-----------------|---------------|---------|--|
|            | Description     |               |         |  |
| Operations | <b>register</b> | MembershipDTO | string  |  |

## 2. Member 클래스다이어그램



| MemberEntityController |                            |                     |            |
|------------------------|----------------------------|---------------------|------------|
| Class Description      | Arguments                  |                     |            |
| 구분                     | Name                       | Type                | Visibility |
|                        | Description                |                     |            |
| Attributes             | <b>memberEntityService</b> | MemberEntityService | private    |

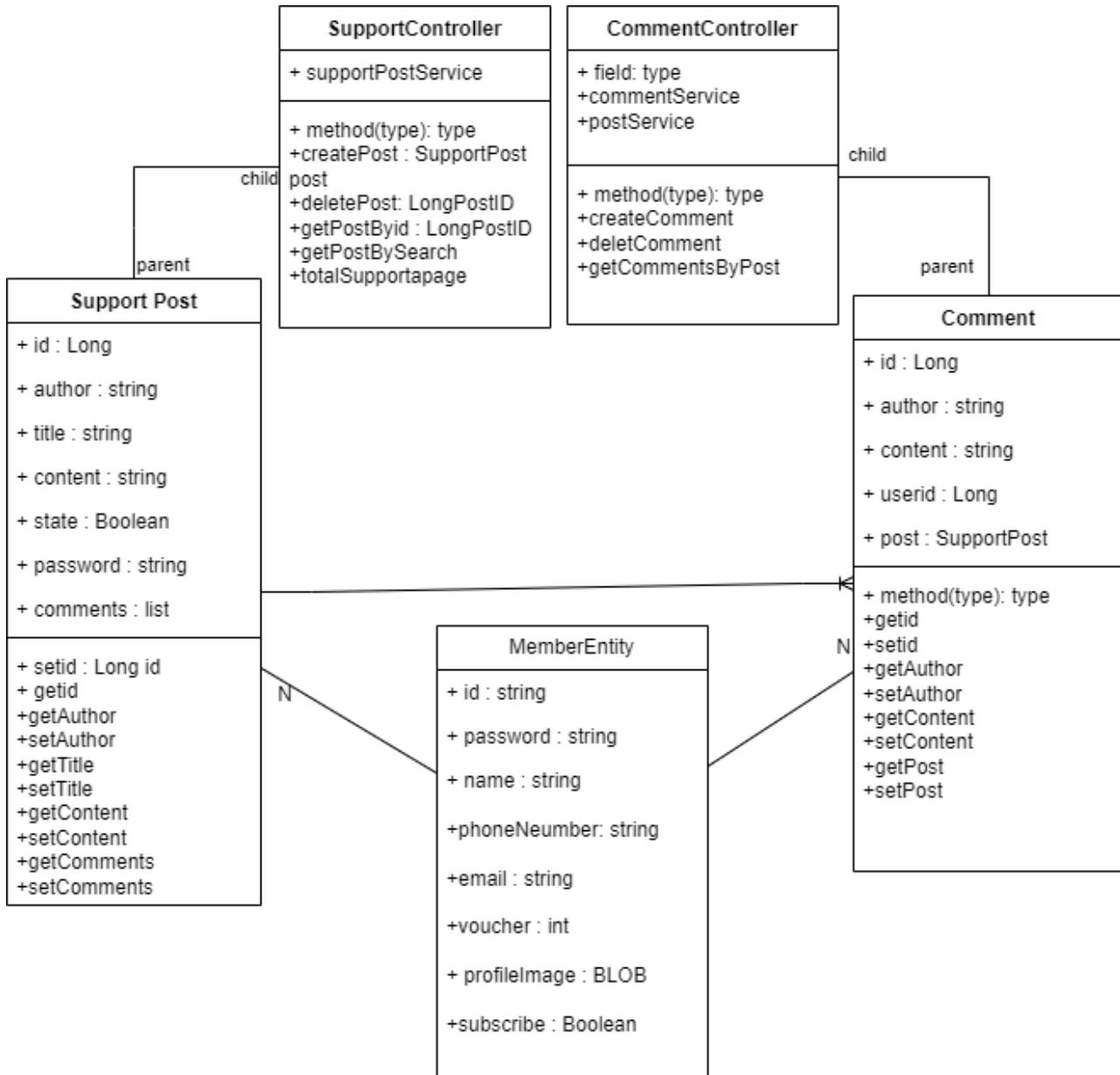
|            |                                   |                                 |                         |
|------------|-----------------------------------|---------------------------------|-------------------------|
|            | <b>loginhistoryService</b>        | LoginhistoryService             | private                 |
| 구분         | Name                              | Argument                        | Returns                 |
|            | Description                       |                                 |                         |
| Operations | <b>getPassword</b>                | String                          | ResponseEntity<String>  |
|            | 임시 비밀번호 전송 성공 여부를 반환.             |                                 |                         |
|            | <b>getSubscribeStatus</b>         | String                          | ResponseEntity<Boolean> |
|            | 사용자 구독 상태(boolean) 반환             |                                 |                         |
|            | <b>register</b>                   | MembershipDTO                   | String                  |
|            | 새 사용자 등록 성공 메시지 반환.               |                                 |                         |
|            | <b>changeNickname</b>             | Map<String, String>             | ResponseEntity<String>  |
|            | 사용자 정보 수정 성공 또는 실패 메시지 반환.        |                                 |                         |
|            | <b>logout</b>                     | HttpServletRequest              | ResponseEntity<String>  |
|            | 세션 종료 후 로그아웃 성공 메시지 반환.           |                                 |                         |
|            | <b>sendTemporaryPassword</b>      | String                          | ResponseEntity<String>  |
|            | 임시 비밀번호 전송 성공 여부를 반환.             |                                 |                         |
|            | <b>getNickname</b>                | String                          | ResponseEntity<String>  |
|            | 사용자 정보 수정 성공 또는 실패 메시지 반환.        |                                 |                         |
|            | <b>login</b>                      | LoginDTO,<br>HttpServletRequest | ResponseEntity<String>  |
|            | 사용자 로그인 및 세션 생성, 성공 시 사용자 아이디 반환. |                                 |                         |
|            | <b>checkUserIdAvailability</b>    | String                          | ResponseEntity<String>  |
|            | 임시 비밀번호 전송 성공 여부를 반환.             |                                 |                         |
|            | <b>reviseUser</b>                 | MembershipDTO                   | String                  |
|            | 사용자 정보 수정 성공 또는 실패 메시지 반환.        |                                 |                         |
|            | <b>subscribe</b>                  | String                          | ResponseEntity<Void>    |
|            | 구독 활성화 성공 상태 반환.                  |                                 |                         |
|            | <b>uploadUserImage</b>            | String, MultipartFile           | ResponseEntity<String>  |
|            | 업로드된 이미지 경로 또는 실패 메시지 반환          |                                 |                         |

|  |                            |                    |                          |
|--|----------------------------|--------------------|--------------------------|
|  | <b>getUserImage</b>        | String             | ResponseEntity<Resource> |
|  | 요청된 사용자의 프로필 이미지 리소스 반환.   |                    |                          |
|  | <b>changePassword</b>      | ChangePasswordDTO  | ResponseEntity<String>   |
|  | 사용자 정보 수정 성공 또는 실패 메시지 반환. |                    |                          |
|  | <b>findUserId</b>          | String, String     | ResponseEntity<String>   |
|  | 이름과 이메일로 아이디를 찾아 반환        |                    |                          |
|  | <b>getLoginStatus</b>      | HttpServletRequest | ResponseEntity<String>   |
|  | 세션에서 로그인된 사용자 아이디 반환       |                    |                          |

| <b>MemberEntityService</b> |   |                        |            |
|----------------------------|---|------------------------|------------|
| Class Description          | 회원 데이터에 대한 비즈니스 로직을 처리하며, 데이터베이스와 상호작용하는 서비스 클래스. |                        |            |
| 구분                         | Name  | Type                   | Visibility |
| Attributes                 | <b>memberEntityRepository</b>                     | memberEntityRepository | private    |
|                            | <b>imageStoragePath</b>                           | Path                   | private    |
| 구분                         | Name  | Argument               | Returns    |
|                            | Description                                       |                        |            |
| Operations                 | <b>authenticateUser</b>                           | LoginDTO               | boolean    |
|                            | 로그인 정보 인증 성공 여부(boolean) 반환.                      |                        |            |
|                            | <b>subscribe</b>                                  | String                 | void       |
|                            | 구독 상태를 활성화  |                        |            |
|                            | <b>loadUserImage</b>                              | String                 | Resource   |
|                            | 요청된 사용자 이미지 리소스 반환.                               |                        |            |
|                            | <b>changeNickname</b>                             | String, String         | boolean    |
|                            | 비밀번호 변경   |                        |            |
|                            | <b>findUserIdByUsernameAndEmail</b>               | String, String         | String     |
|                            | 이름과 이메일로 사용자 아이디를 반환                              |                        |            |
|                            | <b>changePassword</b>                             | ChangePasswordDTO      | boolean    |

|  |   |                       |
|--|---|-----------------------|
|  | 비밀번호 변경                                 |                       |
|  | <b>reviseUser</b>                       | MembershipDTO         |
|  | 사용자 정보 수정                               |                       |
|  | <b>sendTemporaryPassword</b>            | String                |
|  | 임시 비밀번호 전송                              |                       |
|  | <b>getNickname</b>                      | String                |
|  | 요청된 사용자의 닉네임 반환                         |                       |
|  | <b>register</b>                         | MembershipDTO         |
|  | 데이터베이스에 새 사용자 정보를 저장                    |                       |
|  | <b>getPassword</b>                      | String                |
|  | 요청된 사용자의 비밀번호 반환                        |                       |
|  | <b>getMemberById</b>                    | String                |
|  | Optional<MemberEntity>                  |                       |
|  | 요청된 사용자 정보를 Optional<MemberEntity>로 반환. |                       |
|  | <b>isUserIdAvailable</b>                | String                |
|  | 아이디 사용 가능 여부(boolean) 반환.               |                       |
|  | <b>saveUserImage</b>                    | String, MultipartFile |
|  | 이미지 파일 경로 저                             |                       |

### 3. post, comment 클래스다이어그램



| SupportPost       |                       |      |            |
|-------------------|-----------------------|------|------------|
| Class Description | 게시글 정보를 관리하는 엔티티 클래스. |      |            |
| 구분                | Name                  | Type | Visibility |
|                   | Description           |      |            |
| Attributes        | <b>id</b>             | Long | private    |

|            |                        |                           |               |
|------------|------------------------|---------------------------|---------------|
|            | <b>author</b>          | string                    | private       |
|            | <b>title</b>           | string                    | private       |
|            | <b>content</b>         | string                    | private       |
|            | <b>state</b>           | Boolean                   | private       |
|            | <b>password</b>        | password                  | private       |
|            | <b>comments</b>        | List<Comment>             | private       |
| 구분         | Name                   | Argument                  | Returns       |
|            | Description            |                           |               |
| Operations | <b>getId</b>           |                           | Long          |
|            | 게시글 ID를 반환합니다.         |                           |               |
|            | <b>setId</b>           | Long id                   | void          |
|            | <b>getUserid</b>       |                           | String        |
|            | 작성자 이름을 반환합니다.         |                           |               |
|            | <b>setUserId</b>       | String author             | void          |
|            | <b>getTitle</b>        |                           | String        |
|            | <b>setTitle</b>        | String title              | void          |
|            | <b>getContent</b>      |                           | String        |
|            | <b>setContent</b>      | String content            | void          |
|            | <b>getComments</b>     |                           | List<Comment> |
|            | 게시글에 달린 댓글 리스트를 반환합니다. |                           |               |
|            | <b>setComments</b>     | List<Comment><br>comments | void          |
|            | 게시글에 달린 댓글 리스트를 설정합니다. |                           |               |

| SupportPostController |                           |                    |            |
|-----------------------|---------------------------|--------------------|------------|
| Class Description     | 게시글 관련 요청을 처리하는 컨트롤러 클래스. |                    |            |
| 구분                    | Name                      | Type               | Visibility |
|                       | Description               |                    |            |
| Attributes            | <b>supportPostService</b> | SupportPostService | private    |
| 구분                    | Name                      | Argument           | Returns    |
|                       | Description               |                    |            |

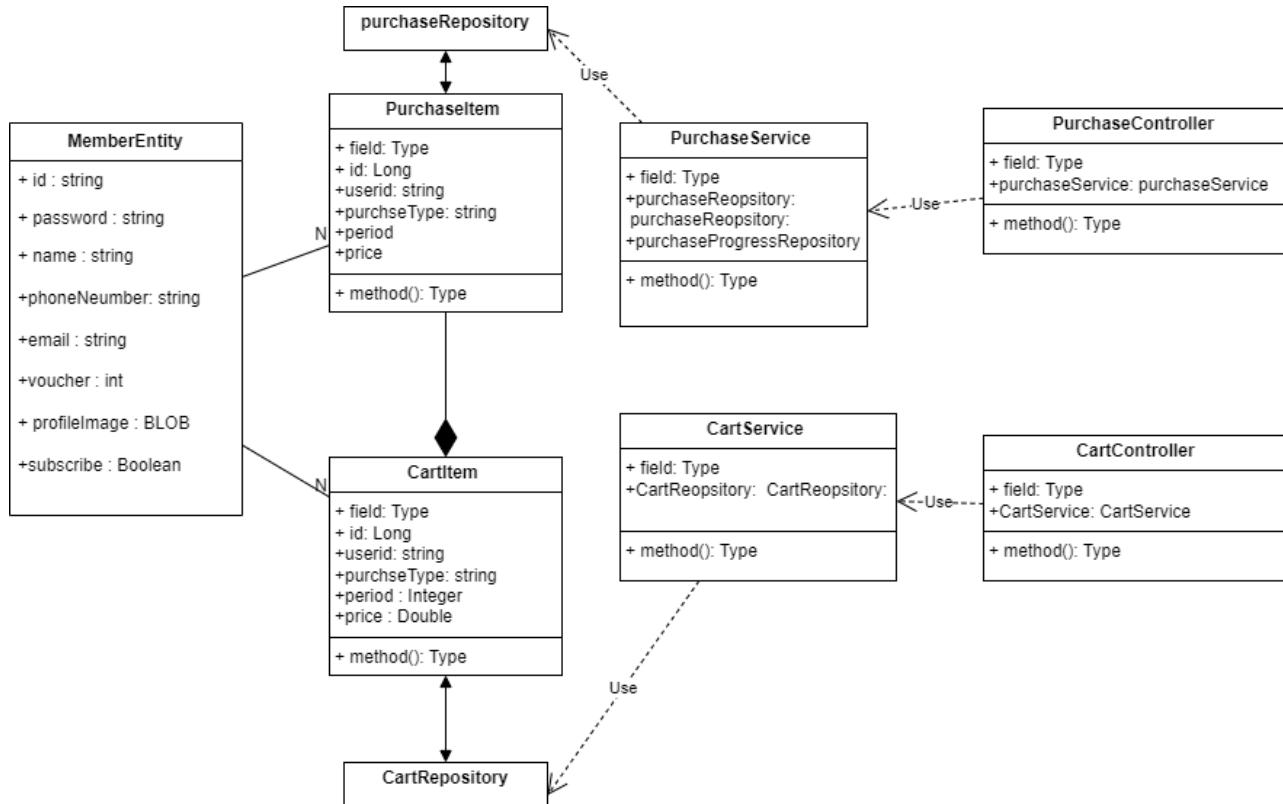
|                          |                        |                          |                                   |
|--------------------------|------------------------|--------------------------|-----------------------------------|
| Operations               | <b>createPost</b>      | SupportPost post         | ResponseEntity<SupportPost>       |
|                          | 새로운 게시글을 작성합니다.        |                          |                                   |
|                          | <b>deletePost</b>      | Long postid              | ResponseEntity<Void>              |
|                          | 특정 게시글을 삭제합니다.         |                          |                                   |
|                          | <b>getPostById</b>     | Long postid              | ResponseEntity<SupportPost>       |
|                          | 특정 게시글을 조회합니다.         |                          |                                   |
|                          | <b>getPostBySearch</b> | String search, int start | ResponseEntity<List<SupportPost>> |
| 검색 조건에 따른 게시글 목록을 반환합니다. |                        |                          |                                   |
| <b>totalSupportpage</b>  |                        | String search            | ResponseEntity<List<Integer>>     |
| 게시글 총 페이지 수를 반환합니다.      |                        |                          |                                   |

| Comment           |                      |             |            |
|-------------------|----------------------|-------------|------------|
| Class Description | 댓글 정보를 관리하는 엔티티 클래스. |             |            |
| 구분                | Name                 | Type        | Visibility |
|                   | Description          |             |            |
| Attributes        | <b>id</b>            | Long        | private    |
|                   | <b>author</b>        | string      | private    |
|                   | <b>content</b>       | string      | private    |
|                   | <b>userid</b>        | Long        | private    |
|                   | <b>post</b>          | SupportPost | private    |
| 구분                | Name                 | Argument    | Returns    |
|                   | Description          |             |            |
| Operations        | <b>getId</b>         |             | Long       |
|                   | 댓글 ID를 반환합니다.        |             |            |
|                   | <b>setId</b>         | Long id     | void       |
|                   | 댓글 ID를 설정합니다         |             |            |
|                   | <b>getAuthor</b>     |             | String     |

|  |                       |                  |             |
|--|-----------------------|------------------|-------------|
|  | 댓글 작성자 이름을 반환합니다.     |                  |             |
|  | <b>setAuthor</b>      | String author    | void        |
|  | 댓글 작성자 이름을 설정합니다.     |                  |             |
|  | <b>getContent</b>     |                  | String      |
|  | <b>getPost</b>        |                  | SupportPost |
|  | <b>setPost</b>        | SupportPost post | void        |
|  | 댓글이 달린 게시글 객체를 설정합니다. |                  |             |

| <b>CommentController</b> |                               |                           |                               |
|--------------------------|-------------------------------|---------------------------|-------------------------------|
| Class Description        | 댓글 관련 요청을 처리하는 컨트롤러 클래스.      |                           |                               |
| 구분                       | Name                          | Type                      | Visibility                    |
|                          | Description                   |                           |                               |
| Attributes               | <b>commentService</b>         | CommentService            | private                       |
|                          | <b>postService</b>            | SupportPostService        | private                       |
| 구분                       | Name                          | Argument                  | Returns                       |
|                          | Description                   |                           |                               |
| Operations               | <b>createComment</b>          | Comment comment           | ResponseEntity<Comment>       |
|                          | 새로운 댓글을 작성합니다.                |                           |                               |
|                          | <b>deleteComment</b>          | Long commentid            | ResponseEntity<Void>          |
|                          | 특정 댓글을 삭제합니다.                 |                           |                               |
|                          | <b>getCommentsByPost</b>      | Long postid , Long userid | ResponseEntity<List<Comment>> |
|                          | 특정 게시글과 사용자에 따른 댓글 목록을 반환합니다. |                           |                               |

#### 4. purchase, cart 클래스



## Purchaseltem

|                   |                       |          |            |
|-------------------|-----------------------|----------|------------|
| Class Description | 게시글 정보를 관리하는 엔티티 클래스. |          |            |
| 구분                | Name                  | Type     | Visibility |
| Description       |                       |          |            |
| Attributes        | <b>id</b>             | Long     | private    |
|                   | <b>userid</b>         | string   | private    |
|                   | <b>purchaseType</b>   | string   | private    |
|                   | <b>period</b>         | string   | private    |
|                   | <b>price</b>          | Double   | private    |
| 구분                | Name                  | Argument | Returns    |
| Description       |                       |          |            |
| Operations        | <b>getId</b>          |          | Long       |
|                   | <b>setId</b>          | Long id  | void       |

|                   |                        |                     |         |
|-------------------|------------------------|---------------------|---------|
|                   | <b>getUserid</b>       |                     | String  |
|                   | <b>setUserId</b>       | String author       | void    |
|                   | <b>getPurchaseType</b> |                     | String  |
|                   | <b>setPurchaseType</b> | String purchaseType | void    |
|                   | 구매 유형 ( 소장/ 대여) 설정합니다  |                     |         |
|                   | <b>getPeriod</b>       |                     | Integer |
|                   | <b>setPeriod</b>       | IntegerPeriod       | void    |
|                   | <b>getPrice</b>        |                     | Double  |
|                   | 구매 항목의 가격을 반환합니다.      |                     |         |
|                   | <b>setPrice</b>        | DoublePrice         | void    |
| 구매 항목의 가격을 설정합니다. |                        |                     |         |

| <b>CartItem</b>   |                           |                     |            |
|-------------------|---------------------------|---------------------|------------|
| Class Description | 장바구니 항목 정보를 저장하는 엔티티 클래스. |                     |            |
| 구분                | Name                      | Type                | Visibility |
|                   | Description               |                     |            |
| Attributes        | <b>id</b>                 | Long                | private    |
|                   | <b>userid</b>             | string              | private    |
|                   | <b>purchaseType</b>       | string              | private    |
|                   | <b>period</b>             | string              | private    |
|                   | <b>price</b>              | Double              | private    |
| 구분                | Name                      | Argument            | Returns    |
|                   | Description               |                     |            |
| Operations        | <b>getId</b>              |                     | Long       |
|                   | <b>setId</b>              | Long id             | void       |
|                   | <b>getUserid</b>          |                     | String     |
|                   | <b>setUserId</b>          | String author       | void       |
|                   | <b>getPurchaseType</b>    |                     | String     |
|                   | <b>setPurchaseType</b>    | String purchaseType | void       |
|                   | 장바구니 항목의 구매 유형을 설정합니다.    |                     |            |

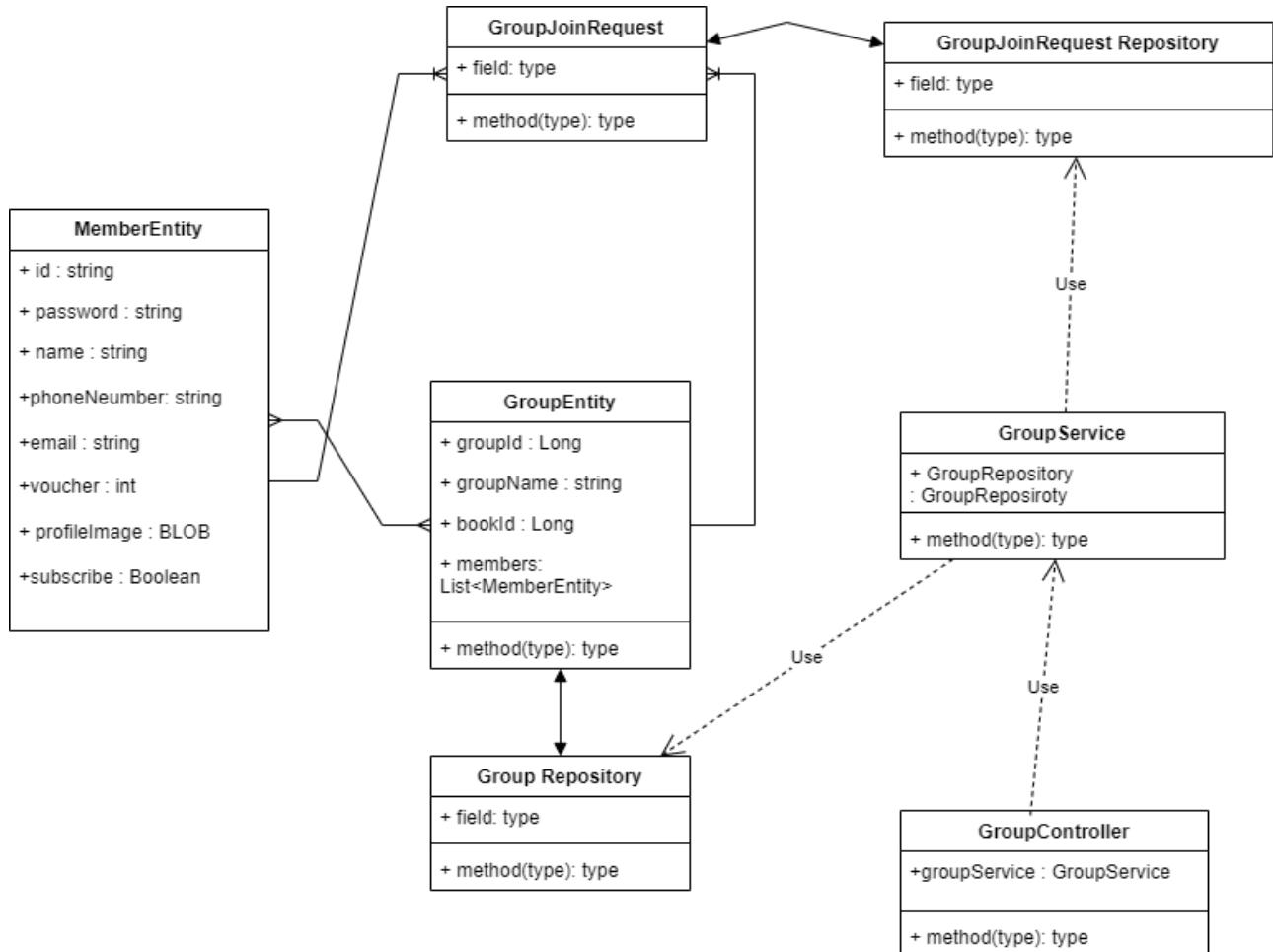
|                     |                      |                |
|---------------------|----------------------|----------------|
| <b>getPeriod</b>    |                      | <b>Integer</b> |
| <b>setPeriod</b>    | <b>IntegerPeriod</b> | <b>void</b>    |
| <b>getPrice</b>     |                      | <b>Double</b>  |
| 장바구니 항목의 가격을 반환합니다. |                      |                |
| <b>setPrice</b>     | <b>DoublePrice</b>   | <b>void</b>    |
| 장바구니 항목의 가격을 설정합니다. |                      |                |

| <b>PurchaseController</b> |                                   |                              |                                  |
|---------------------------|-----------------------------------|------------------------------|----------------------------------|
| <b>Class Description</b>  | 구매 관련 요청을 처리하는 컨트롤러 클래스.          |                              |                                  |
| <b>구분</b>                 | Name                              | Type                         | Visibility                       |
|                           | Description                       |                              |                                  |
| <b>Attributes</b>         | <b>purchaseService</b>            | PurchaseService              | private                          |
| <b>구분</b>                 | Name                              | Argument                     | Returns                          |
|                           | Description                       |                              |                                  |
| <b>Operations</b>         | <b>addPurchaseItem</b>            | PurchaseItem<br>purchaseItem | ResponseEntity<<br>PurchaseItem> |
|                           | 구매 항목을 추가합니다.                     |                              |                                  |
|                           | <b>getPurchaseItemByIdAndUser</b> | Long id, String userid       | ResponseEntity<<br>PurchaseItem> |
|                           | 특정 유저의 특정 구매 항목을 조회합니다.           |                              |                                  |
|                           | <b>removePurchaseItem</b>         | String userid, Long itemId   | ResponseEntity<<br>Void>         |
|                           | 특정 유저의 특정 구매 항목을 삭제합니다.           |                              |                                  |

| <b>CartController</b>    |                            |      |            |
|--------------------------|----------------------------|------|------------|
| <b>Class Description</b> | 장바구니 관련 요청을 처리하는 컨트롤러 클래스. |      |            |
| <b>구분</b>                | Name                       | Type | Visibility |

|            | Description                 |                            |                        |
|------------|-----------------------------|----------------------------|------------------------|
| Attributes | <b>cartService</b>          | CartService                | private                |
| 구분         | Name                        | Argument                   | Returns                |
|            | Description                 |                            |                        |
| Operations | <b>addBookToCart</b>        | CartItem                   | ResponseType<CartItem> |
|            | 장바구니에 책을 추가합니다.             |                            |                        |
|            | <b>getCartItemsByUserId</b> | String userid              | ResponseType<CartItem> |
|            | 특정 유저의 장바구니 항목을 조회합니다.      |                            |                        |
|            | <b>removeBookFromCart</b>   | String userid, Long itemId | ResponseType<Void>     |
|            | 특정 유저의 특정 장바구니 항목을 삭제합니다.   |                            |                        |

## 5. 그룹 가입, 그룹 요청 클래스



| Group Entity      |                                 |         |            |
|-------------------|---------------------------------|---------|------------|
| Class Description | 그룹 정보를 저장하는 엔티티 클래스.            |         |            |
| 구분                | Name                            | Type    | Visibility |
|                   | Description                     |         |            |
| Attributes        | groupId                         | Long    | private    |
|                   | groupName                       | string  | private    |
|                   | bookId                          | string  | private    |
|                   | startDate                       | string  | private    |
|                   | endDate                         | Double  | private    |
|                   | state                           | Integer | private    |
|                   | 그룹 상태 (0: 시작 전, 1: 진행 중, 2: 완료) |         |            |
|                   | leaderId                        | String  | private    |
|                   | 그룹리더ID                          |         |            |

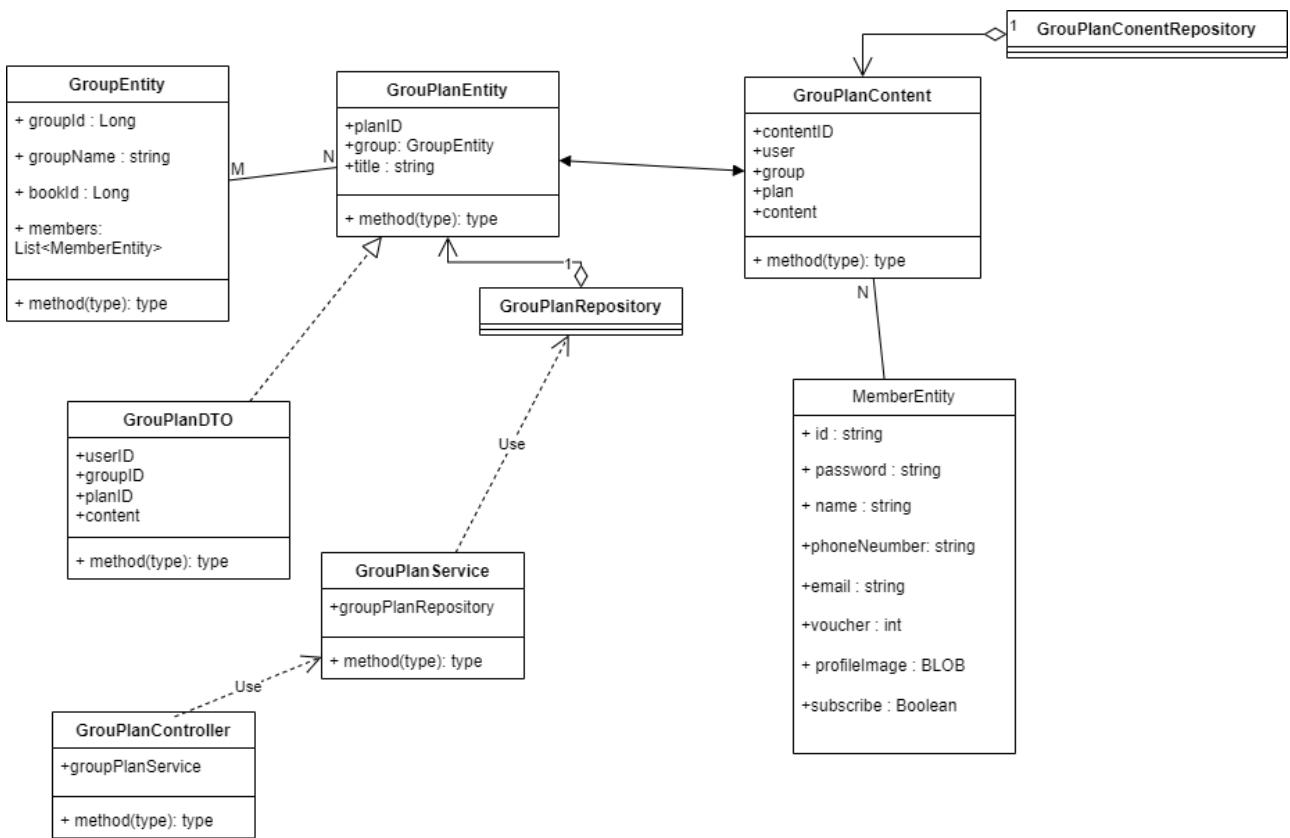
|            |                     |                               |                    |
|------------|---------------------|-------------------------------|--------------------|
|            | <b>authority</b>    | String                        | private            |
| 리더의 권한     |                     |                               |                    |
|            | <b>members</b>      | List<MemberEntity>            | private            |
| 구분         | Name                | Argument                      | Returns            |
|            | Description         |                               |                    |
| Operations | <b>getGroupName</b> |                               | String             |
|            | <b>setGroupName</b> | String group Name             | void               |
|            | <b>getMembers</b>   |                               | List<MemberEntity> |
|            | <b>setMembers</b>   | List<MemberEntity><br>members | void               |

| <b>GroupJoinRequest</b> |                            |                   |              |
|-------------------------|----------------------------|-------------------|--------------|
| Class Description       | 그룹 가입 요청 정보를 저장하는 엔티티 클래스. |                   |              |
| 구분                      | Name                       | Type              | Visibility   |
|                         | Description                |                   |              |
| Attributes              | <b>requestId</b>           | Long              | private      |
|                         | <b>user</b>                | MemberEntity      | private      |
| 가입 요청을 보낸 사용자           |                            |                   |              |
| Attributes              | <b>group</b>               | GroupEntity       | private      |
|                         | 가입 요청 대상 그룹                |                   |              |
| 구분                      | Name                       | Argument          | Returns      |
|                         | Description                |                   |              |
| Operations              | <b>getUser</b>             |                   | MemberEntity |
|                         | 가입 요청 사용자 반환               |                   |              |
|                         | <b>setUser</b>             | MemberEntity User | void         |
|                         | <b>getGroup</b>            |                   | GroupEntity  |
| Operations              | <b>setGroup</b>            | GroupEntity group | void         |

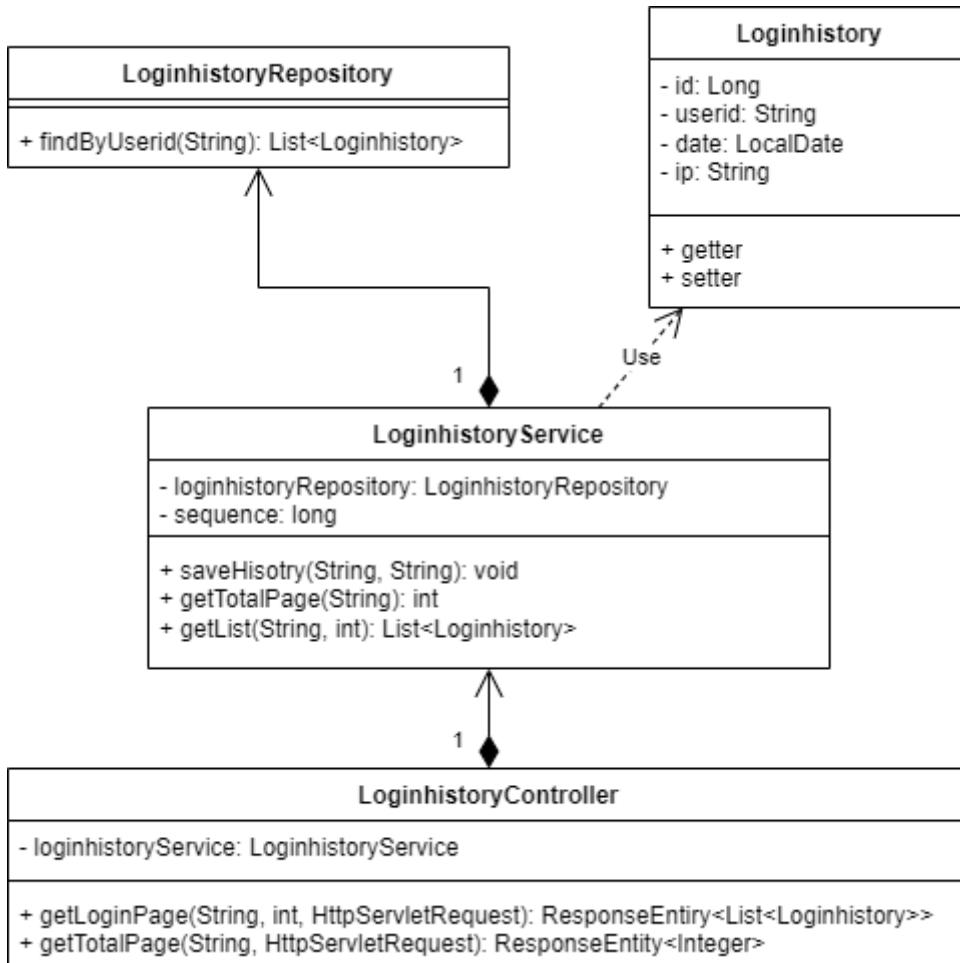
| GroupService      |                                   |                                 |                    |
|-------------------|-----------------------------------|---------------------------------|--------------------|
| Class Description | 그룹 관련 비즈니스 로직을 처리하는 서비스 클래스.      |                                 |                    |
| 구분                | Name                              | Type                            | Visibility         |
|                   | Description                       |                                 |                    |
| Attributes        | <b>groupRepository</b>            | GroupRepository                 | private            |
|                   | 그룹 데이터 접근                         |                                 |                    |
|                   | <b>memberRepository</b>           | MemberEntityRepository          | private            |
| Operations        | <b>groupJoinRequestRepository</b> | GroupJoinRequestRepository      | private            |
|                   | Name                              | Argument                        | Returns            |
|                   | Description                       |                                 |                    |
| Operations        | <b>createGroup</b>                | group DTO                       | void               |
|                   | 새로운 그룹을 생성                        |                                 |                    |
|                   | <b>getGroupsByState</b>           | Integer State                   | List<groupDTO>     |
|                   | 상태에 따라 그룹 리스트 반환                  |                                 |                    |
|                   | <b>joinGroup</b>                  | String userId, Long groupId     | void               |
|                   | 사용자를 그룹에 추가                       |                                 |                    |
|                   | <b>getGroupDetail</b>             | Long groupId                    | GroupDetailDTO     |
|                   | 특정 그룹의 세부 정보 반환                   |                                 |                    |
|                   | <b>reviseGroup</b>                | Long groupId, GroupDTO groupDTO | void               |
|                   | 그룹 정보를 수정                         |                                 |                    |
| Operations        | <b>requestJoinGroup</b>           | String userId, Long groupId     | void               |
|                   | 그룹가입 요청생                          |                                 |                    |
|                   | <b>getJoinRequest</b>             | Long groupId                    | List<MemberEntity> |
|                   | 특정 그룹의 가입 요청 반환                   |                                 |                    |
|                   | <b>rejectJoinRequest</b>          | String userid, Long group id    | void               |
| 특정 가입 요청을 거절      |                                   |                                 |                    |

| GroupController   |                          |                                    |                        |
|-------------------|--------------------------|------------------------------------|------------------------|
| Class Description | 그룹 관련 요청을 처리하는 컨트롤러 클래스. |                                    |                        |
| 구분                | Name                     | Type                               | Visibility             |
|                   | Description              |                                    |                        |
| Attributes        | <b>groupService</b>      | GroupService                       | private                |
| 구분                | Name                     | Argument                           | Returns                |
|                   | Description              |                                    |                        |
| Operations        | <b>createGroup</b>       | group DTO                          | void                   |
|                   | 새로운 그룹을 생성               |                                    |                        |
|                   | <b>getGroupsByState</b>  | Integer State                      | List<groupDTO>         |
|                   | 상태에 따라 그룹 리스트 반환         |                                    |                        |
|                   | <b>getGroupDetail</b>    | Long groupId                       | GroupDetailDTO         |
|                   | 특정 그룹의 세부 정보 반환          |                                    |                        |
|                   | <b>reviseGroup</b>       | Long groupId,<br>GroupDTO groupDTO | void                   |
|                   | 그룹 정보를 수정                |                                    |                        |
|                   | <b>requestJoinGroup</b>  | String userId, Long<br>groupId     | void                   |
|                   | 그룹가입 요청생                 |                                    |                        |
|                   | <b>getJoinRequest</b>    | Long groupId                       | List<MemberEntit<br>y> |
|                   | 특정 그룹의 가입 요청 반환          |                                    |                        |
|                   | <b>rejectJoinRequest</b> | string userid , Long<br>group id   | void                   |
|                   | 특정 가입 요청을 거절             |                                    |                        |

## 6. 그룹플랜 다이어그램



## 7. 로그인 기록 클래스 다이어그램



| LoginhistoryRepository |   |          |                    |
|------------------------|---|----------|--------------------|
| Class Description      | 로그인 기록 데이터를 데이터베이스에서 조회하거나 저장하는 JPA Repository. |          |                    |
| 구분                     | Name  | Argument | Returns            |
|                        | Description                                     |          |                    |
| Operations             | findByUserId                                    | String   | List<Loginhistory> |
|                        | 특정 사용자의 모든 로그인 기록을 반환.                          |          |                    |

| Loginhistory      |                            |      |            |
|-------------------|----------------------------|------|------------|
| Class Description | 사용자의 로그인 기록을 저장하는 JPA 엔티티. |      |            |
| 구분                | Name                       | Type | Visibility |
|                   | Description                |      |            |
| Attributes        | id                         | Long | private    |

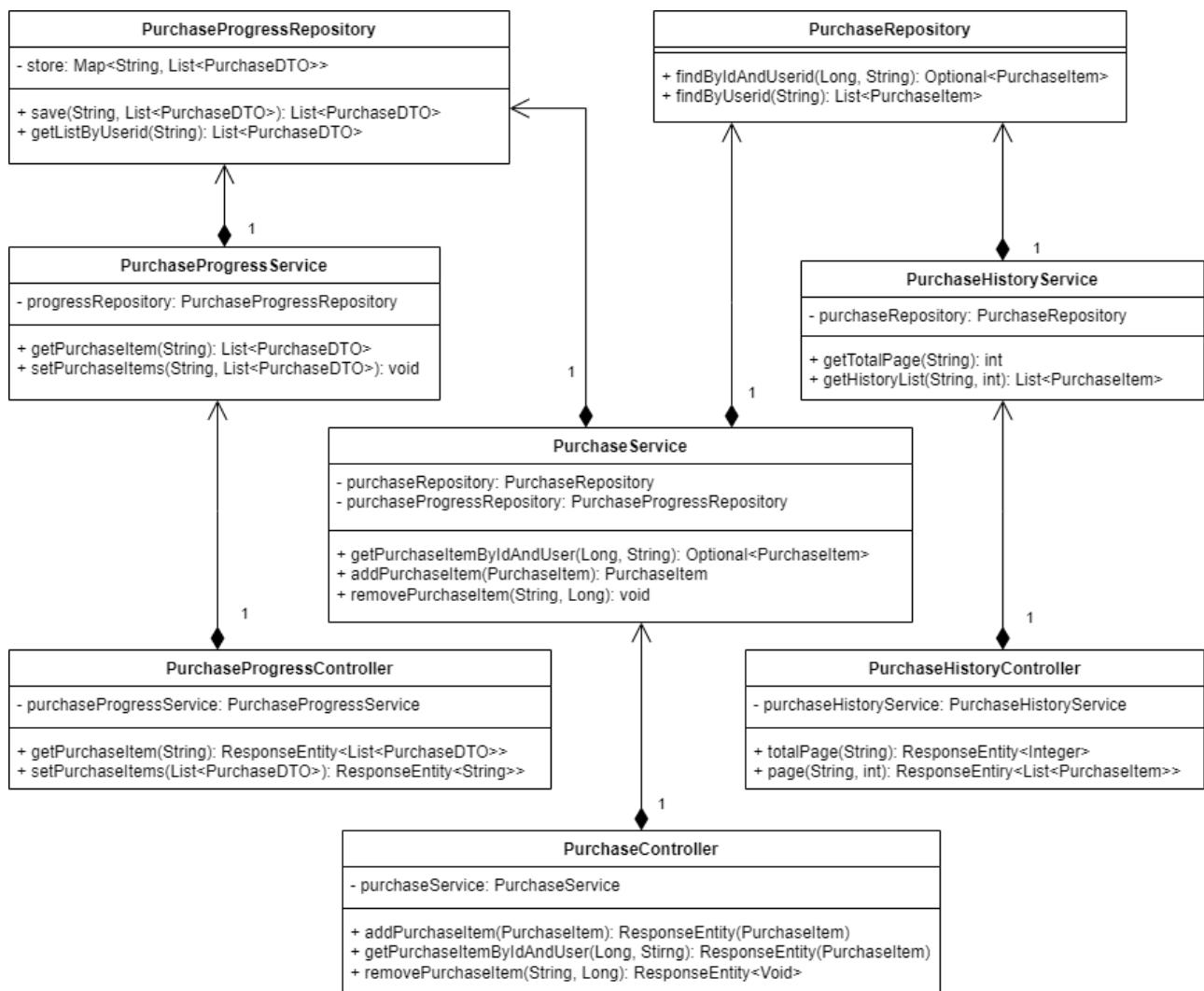
|            |                            |           |         |
|------------|----------------------------|-----------|---------|
|            | 로그인 기록 레코드의 고유의 아이디        |           |         |
|            | <b>userid</b>              | String    | private |
|            | <b>date</b>                | LocalDate | private |
|            | <b>ip</b>                  | String    | private |
| 구분         | Name                       | Argument  | Returns |
|            | Description                |           |         |
| Operations | <b>getter &amp; setter</b> |           |         |

| <b>LoginhistoryService</b> |                                  |                        |                    |
|----------------------------|----------------------------------|------------------------|--------------------|
| Class Description          | 로그인 기록 관련 비즈니스 로직을 처리하는 서비스 클래스. |                        |                    |
| 구분                         | Name                             | Type                   | Visibility         |
|                            | Description                      |                        |                    |
| Attributes                 | <b>loginhistoryRepository</b>    | LoginhistoryRepository | private            |
|                            | <b>sequence</b>                  | long                   | private            |
| 구분                         | Name                             | Argument               | Returns            |
|                            | Description                      |                        |                    |
| Operations                 | <b>saveHistory</b>               | String, String         | void               |
|                            | 사용자의 로그인 정보를 저장소에 기록.            |                        |                    |
|                            | <b>getTotalPage</b>              | String                 | int                |
|                            | 주어진 사용자에 대한 로그인 기록의 총 페이지 수 계산.  |                        |                    |
|                            | <b>getList</b>                   | String, int            | List<Loginhistory> |
|                            | 주어진 사용자의 특정 페이지의 로그인 기록 리스트 반환.  |                        |                    |

| <b>LoginhistoryController</b> |                               |                     |            |
|-------------------------------|-------------------------------|---------------------|------------|
| Class Description             | 로그인 기록에 대한 API 요청을 처리하는 컨트롤러. |                     |            |
| 구분                            | Name                          | Type                | Visibility |
|                               | Description                   |                     |            |
| Attributes                    | <b>loginhistoryService</b>    | LoginhistoryService | private    |
| 구분                            | Name                          | Argument            | Returns    |
|                               | Description                   |                     |            |

|            |                     |                               |  |
|------------|---------------------|-------------------------------|--|
| Operations | <b>getLoginPage</b> | String,<br>HttpServletRequest | ResponseEntity<br><List<Loginhistory<br>>> |
|            |                     |                               | 사용자의 로그인 기록 리스트를 요청한 페이지 단위로 반환.           |
|            | <b>getTotalPage</b> | String,<br>HttpServletRequest | ResponseEntity<br><Integer>                |
|            |                     |                               | 사용자의 로그인 기록 총 페이지 수를 반환.                   |

## 8. 구매/대여 클래스 다이어그램



**PurchaseProgressRepository**

|                           |  |                                |                       |
|---------------------------|--|--------------------------------|-----------------------|
| Class Description         | 구매 진행 정보를 메모리 기반 저장소에 관리하는 Repository. |                                |                       |
| 구분                        | Name                                   | Type                           | Visibility            |
|                           | Description                            |                                |                       |
| Attributes                | <b>store</b>                           | Map<String, List<PurchaseDTO>> | private               |
| 구분                        | Name                                   | Argument                       | Returns               |
|                           | Description                            |                                |                       |
| Operations                | <b>save</b>                            | String,<br>List<PurchaseDTO>   | List<br><PurchaseDTO> |
|                           | 사용자의 기존 구매 데이터를 새로운 리스트로 덮어씀.          |                                |                       |
|                           | <b>getListByUserId</b>                 | String                         | List<br><PurchaseDTO> |
| 사용자 ID로 구매 진행 내역 리스트를 조회. |  |                                |                       |

| <b>PurchaseProgressService</b> |                                       |                              |                       |
|--------------------------------|---------------------------------------|------------------------------|-----------------------|
| Class Description              | 구매 진행 정보를 저장 및 조회하는 로직을 처리하는 서비스 클래스. |                              |                       |
| 구분                             | Name                                  | Type                         | Visibility            |
|                                | Description                           |                              |                       |
| Attributes                     | <b>progressRepository</b>             | PurchaseProgressRepository   | private               |
| 구분                             | Name                                  | Argument                     | Returns               |
|                                | Description                           |                              |                       |
| Operations                     | <b>getPurchaseItem</b>                | String                       | List<br><PurchaseDTO> |
|                                | 사용자 ID로 구매 진행 내역을 조회.                 |                              |                       |
|                                | <b>setPurchaseItems</b>               | String,<br>List<PurchaseDTO> | void                  |
| 사용자 ID로 새로운 구매 데이터를 저장.        |                                       |                              |                       |

| <b>PurchaseProgressController</b> |                                  |      |            |
|-----------------------------------|----------------------------------|------|------------|
| Class Description                 | 구매 진행 정보와 관련된 API 요청을 처리하는 컨트롤러. |      |            |
| 구분                                | Name                             | Type | Visibility |

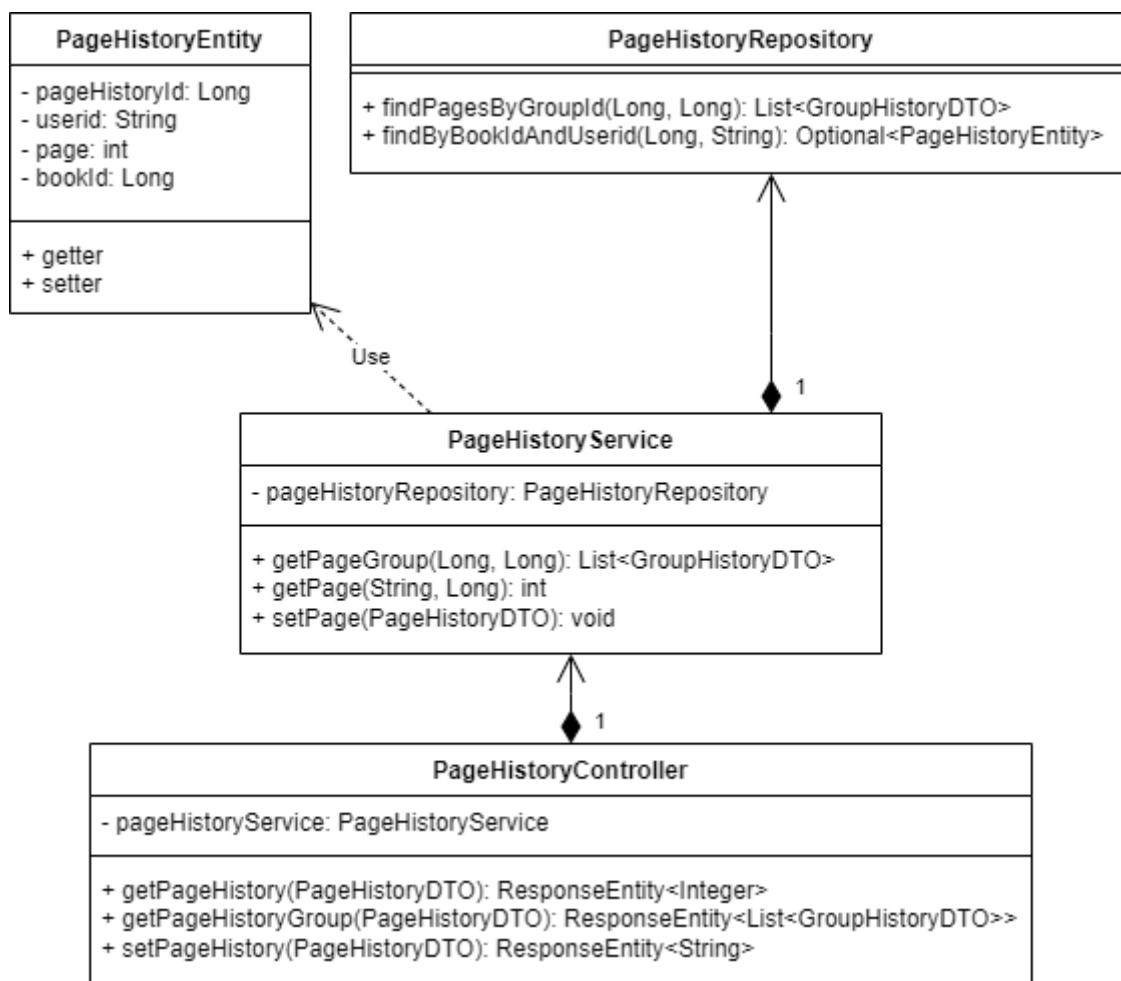
|                    | Description                    |                         |                                   |
|--------------------|--------------------------------|-------------------------|-----------------------------------|
| Attributes         | <b>purchaseProgressService</b> | PurchaseProgressService | private                           |
| 구분                 | Name                           | Argument                | Returns                           |
|                    | Description                    |                         |                                   |
| Operations         | <b>getPurchaseItem</b>         | String                  | ResponseEntity<List<PurchaseDTO>> |
|                    | 사용자의 구매 진행 내역을 반환.             |                         |                                   |
|                    | <b>setPurchaseItems</b>        | List<PurchaseDTO>       | ResponseEntity<String>            |
| 사용자의 구매 진행 내역을 저장. |                                |                         |                                   |

| <b>PurchaseHistoryService</b>  |   |                    |                    |
|--------------------------------|---|--------------------|--------------------|
| Class Description              | 구매 내역의 총 페이지 계산 및 내역 조회 로직을 처리하는 서비스 클래스. |                    |                    |
| 구분                             | Name                                      | Type               | Visibility         |
|                                | Description                               |                    |                    |
| Attributes                     | <b>purchaseRepository</b>                 | PurchaseRepository | private            |
| 구분                             | Name                                      | Argument           | Returns            |
| Operations                     | <b>getTotalPage</b>                       | String             | int                |
|                                | 주어진 사용자에 대한 구매 내역 총 페이지 수를 계산.            |                    |                    |
|                                | <b>getHistoryList</b>                     | String, int        | List<PurchaseItem> |
| 주어진 사용자에 대한 특정 페이지의 구매 내역을 반환. |   |                    |                    |

| <b>PurchaseHistoryController</b> |                               |                        |            |
|----------------------------------|-------------------------------|------------------------|------------|
| Class Description                | 구매 내역과 관련된 API 요청을 처리하는 컨트롤러. |                        |            |
| 구분                               | Name                          | Type                   | Visibility |
|                                  | Description                   |                        |            |
| Attributes                       | <b>purchaseHistoryService</b> | PurchaseHistoryService | private    |

| 구분                          | Name                    | Argument    | Returns                            |
|-----------------------------|-------------------------|-------------|------------------------------------|
| Description                 |                         |             |                                    |
| Operations                  | <b>totalPage</b>        | String      | ResponseEntity<Integer>            |
|                             | 사용자의 구매 내역 총 페이지 수를 반환. |             |                                    |
|                             | <b>page</b>             | String, int | ResponseEntity<List<PurchaseItem>> |
| 사용자의 구매 내역을 요청한 페이지 단위로 반환. |                         |             |                                    |

## 9. 페이지 기록 클래스 다이어그램



**PageHistoryEntity**

|                   |                             |          |            |
|-------------------|-----------------------------|----------|------------|
| Class Description | 사용자의 책 열람 기록을 저장하는 JPA 엔티티. |          |            |
| 구분                | Name                        | Type     | Visibility |
|                   | Description                 |          |            |
| Attributes        | <b>pageHistoryId</b>        | Long     | private    |
|                   | <b>userid</b>               | String   | private    |
|                   | <b>page</b>                 | int      | private    |
|                   | <b>bookId</b>               | Long     | private    |
| 구분                | Name                        | Argument | Returns    |
|                   | Description                 |          |            |
| Operations        | <b>getter &amp; setter</b>  |          |            |

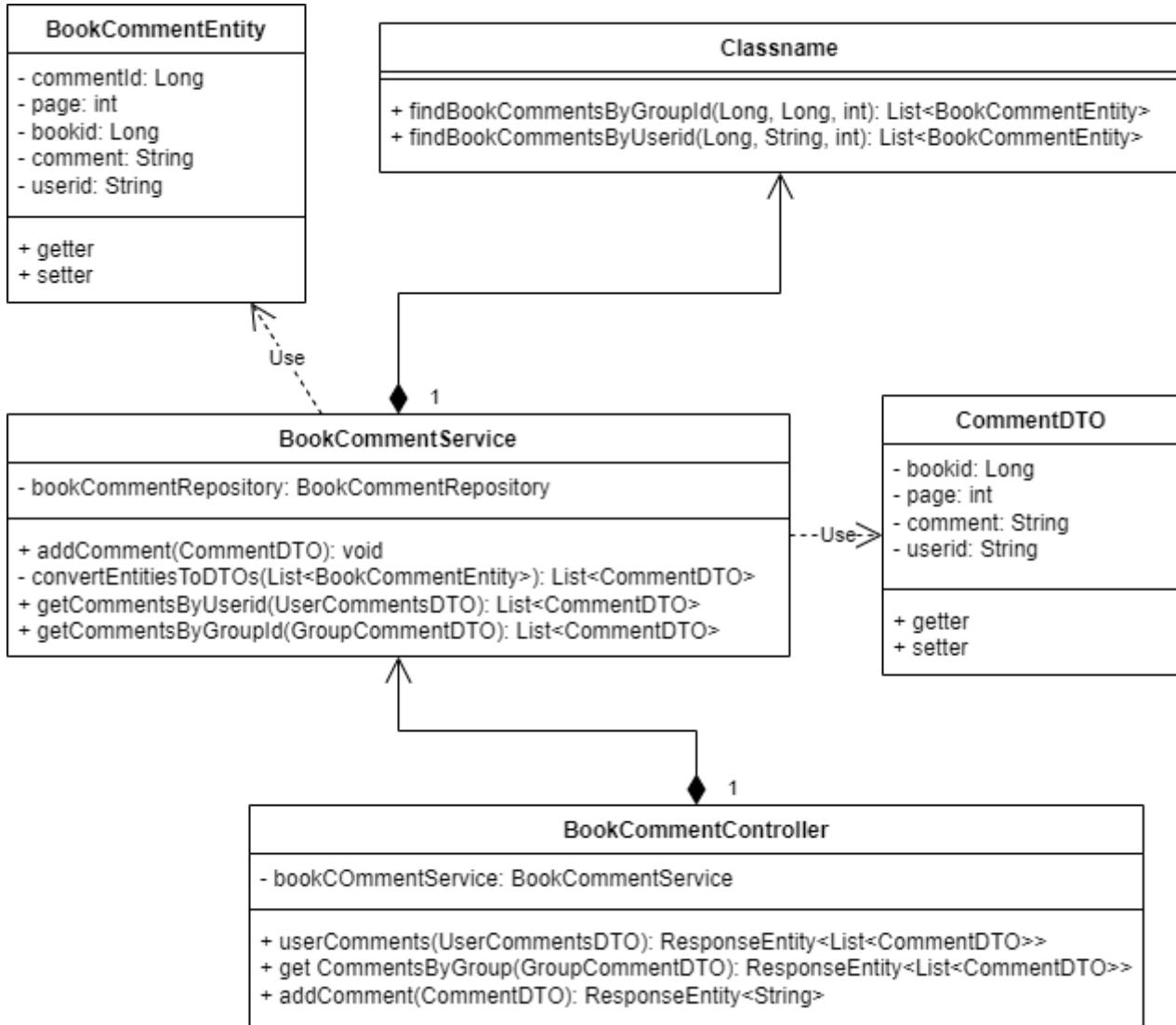
| <b>PageHistoryRepository</b>  |                                   |              |                              |
|-------------------------------|-----------------------------------|--------------|------------------------------|
| Class Description             | 책 열람 기록 데이터를 관리하는 JPA Repository. |              |                              |
| 구분                            | Name                              | Argument     | Returns                      |
|                               | Description                       |              |                              |
| Operations                    | <b>findPagesByGroupId</b>         | Long, Long   | List<GroupHistory DTO>       |
|                               | 그룹 ID와 책 ID로 그룹의 열람 기록 리스트를 반환.   |              |                              |
|                               | <b>findByBookIdAndUserId</b>      | Long, String | Optional<Page HistoryEntity> |
| 특정 사용자와 책 ID에 해당하는 열람 기록을 반환. |                                   |              |                              |

| <b>PageHistoryService</b> |   |                       |            |
|---------------------------|---|-----------------------|------------|
| Class Description         | 책 열람 기록을 저장 및 조회하는 비즈니스 로직을 처리하는 서비스 클래스. |                       |            |
| 구분                        | Name                                      | Type                  | Visibility |
|                           | Description                               |                       |            |
| Attributes                | <b>pageHistoryRepository</b>              | PageHistoryRepository | private    |
| 구분                        | Name                                      | Argument              | Returns    |
|                           | Description                               |                       |            |

|                              |                                 |                |                        |
|------------------------------|---------------------------------|----------------|------------------------|
| Operations                   | <b>getPageGroup</b>             | Long, Long     | List<GroupHistory DTO> |
|                              | 그룹 ID와 책 ID로 그룹의 열람 기록을 조회.     |                |                        |
|                              | <b>getPage</b>                  | String, Long   | int                    |
|                              | 사용자와 책 ID로 특정 책의 열람 기록 페이지를 반환. |                |                        |
|                              | <b>setPage</b>                  | PageHistoryDTO | void                   |
| 책의 열람 기록을 저장하거나 기존 기록을 업데이트. |                                 |                |                        |

| <b>PageHistoryController</b> |  |                    |                                       |
|------------------------------|--|--------------------|---------------------------------------|
| Class Description            | 책의 열람 기록 저장 및 조회와 관련된 API 요청을 처리하는 컨트롤러. |                    |                                       |
| 구분                           | Name                                     | Type               | Visibility                            |
|                              | Description                              |                    |                                       |
| Attributes                   | <b>pageHistoryService</b>                | PageHistoryService | private                               |
| 구분                           | Name                                     | Argument           | Returns                               |
|                              | Description                              |                    |                                       |
| Operations                   | <b>getPageHistory</b>                    | PageHistoryDTO     | ResponseType <Integer>                |
|                              | 사용자의 특정 책에 대한 열람 기록 페이지를 반환.             |                    |                                       |
|                              | <b>getPageHistoryGroup</b>               | PageHistoryDTO     | ResponseType <List<Group HistoryDTO>> |
|                              | 그룹 내 특정 책에 대한 열람 기록 리스트를 반환.             |                    |                                       |
|                              | <b>setPageHistory</b>                    | PageHistoryDTO     | ResponseType <String>                 |
| 책 열람 기록을 저장하고 성공 메시지를 반환.    |  |                    |                                       |

## 10. 감상평 클래스 다이어그램



| BookCommentEntity |                                 |          |            |
|-------------------|---------------------------------|----------|------------|
| Class Description | 책에 대한 사용자의 댓글 정보를 저장하는 JPA 엔티티. |          |            |
| 구분                | Name                            | Type     | Visibility |
| Attributes        | commentId                       | Long     | private    |
|                   | page                            | int      | private    |
|                   | bookid                          | Long     | private    |
|                   | comment                         | String   | private    |
|                   | userid                          | String   | private    |
| 구분                | Name                            | Argument | Returns    |
|                   | Description                     |          |            |

|            |                            |  |  |
|------------|----------------------------|--|--|
| Operations | <b>getter &amp; setter</b> |  |  |
|------------|----------------------------|--|--|

| <b>BookCommentRepository</b>          |                                      |                   |                         |
|---------------------------------------|--------------------------------------|-------------------|-------------------------|
| Class Description                     | 책 댓글 데이터를 조회하거나 저장하는 JPA 레포지토리.      |                   |                         |
| 구분                                    | Name                                 | Argument          | Returns                 |
|                                       | Description                          |                   |                         |
| Operations                            | <b>findBookCommentsByGroupid</b>     | Long, Long, int   | List<BookCommentEntity> |
|                                       | 그룹 ID, 책 ID, 페이지를 기준으로 책 댓글 리스트를 반환. |                   |                         |
|                                       | <b>findBookCommentsByUserId</b>      | Long, String, int | List<BookCommentEntity> |
| 사용자 ID, 책 ID, 페이지를 기준으로 책 댓글 리스트를 반환. |                                      |                   |                         |

| <b>CommentDTO</b> |                                 |          |            |
|-------------------|---------------------------------|----------|------------|
| Class Description | 댓글 데이터를 주고받기 위한 데이터 전송 객체(DTO). |          |            |
| 구분                | Name                            | Type     | Visibility |
|                   | Description                     |          |            |
| Attributes        | <b>bookid</b>                   | Long     | private    |
|                   | <b>page</b>                     | int      | private    |
|                   | <b>comment</b>                  | String   | private    |
|                   | <b>userid</b>                   | String   | private    |
| 구분                | Name                            | Argument | Returns    |
|                   | Description                     |          |            |
| Operations        | <b>getter &amp; setter</b>      |          |            |

| <b>BookCommentService</b> |                                |             |            |
|---------------------------|--------------------------------|-------------|------------|
| Class Description         | 책 댓글 추가 및 조회 로직을 처리하는 서비스 클래스. |             |            |
| 구분                        | Name                           | Type        | Visibility |
|                           | Description                    |             |            |
| Attributes                | <b>bookComment</b>             | BookComment | private    |

|            | <b>Repository</b>                   | Repository              |                  |
|------------|-------------------------------------|-------------------------|------------------|
| 구분         | Name                                | Argument                | Returns          |
|            | Description                         |                         |                  |
| Operations | <b>addComment</b>                   | CommentDTO              | void             |
|            | CommentDTO를 받아 책에 대한 댓글을 저장.        |                         |                  |
|            | <b>convertEntitiesToDTOs</b>        | List<BookCommentEntity> | List<CommentDTO> |
|            | 댓글 엔티티 리스트를 DTO 리스트로 변환.            |                         |                  |
|            | <b>getCommentsByUserId</b>          | UserCommentsDTO         | List<CommentDTO> |
|            | 사용자 ID와 책 ID를 기준으로 댓글 리스트를 조회 및 반환. |                         |                  |
|            | <b>getCommentsByGroupId</b>         | GroupCommentDTO         | List<CommentDTO> |
|            | 그룹 ID와 책 ID를 기준으로 댓글 리스트를 조회 및 반환.  |                         |                  |

| <b>BookCommentController</b> |                                    |                    |                                |
|------------------------------|------------------------------------|--------------------|--------------------------------|
| Class Description            |                                    |                    |                                |
| 구분                           | Name                               | Type               | Visibility                     |
|                              | 책 댓글 추가 및 조회 관련 API 요청을 처리하는 컨트롤러. |                    |                                |
| Attributes                   | <b>bookCommentService</b>          | BookCommentService | private                        |
| 구분                           | Name                               | Argument           | Returns                        |
|                              | Description                        |                    |                                |
| Operations                   | <b>userComments</b>                | UserCommentsDTO    | ResponseType<List<CommentDTO>> |
|                              | 특정 사용자의 댓글 리스트를 반환.                |                    |                                |
|                              | <b>getCommentsByGroup</b>          | GroupCommentDTO    | ResponseType<List<CommentDTO>> |
|                              | 특정 그룹의 책에 대한 댓글 리스트를 반환.           |                    |                                |
|                              | <b>addComment</b>                  | CommentDTO         | ResponseType<String>           |
| 책에 새로운 댓글을 추가.               |                                    |                    |                                |

#### 4. Sequence diagram

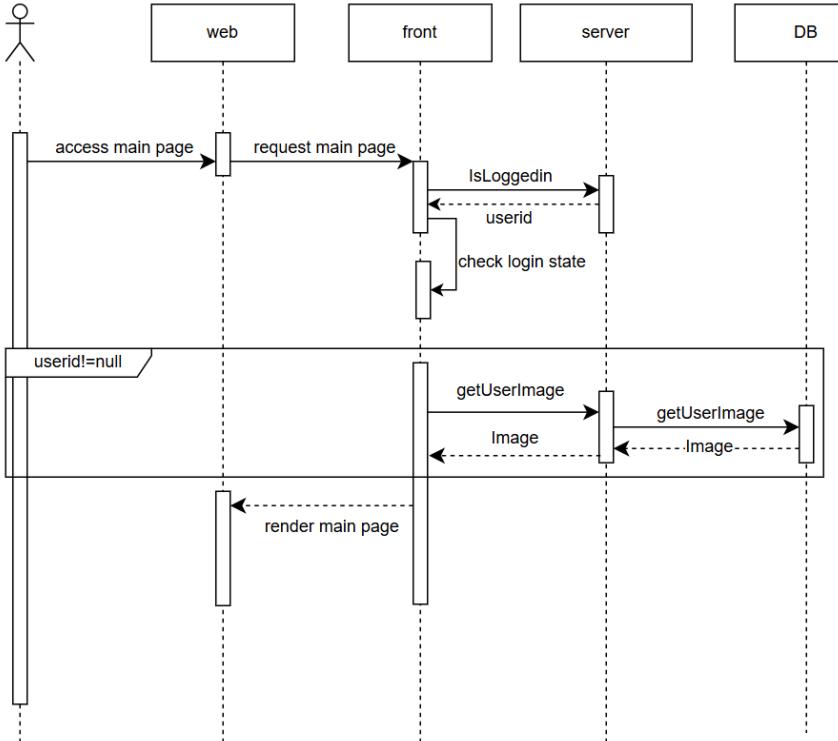
- Draw sequence diagrams for the whole functions of your system.
  - Explain each sequence diagram.
- 12pt, 160%.

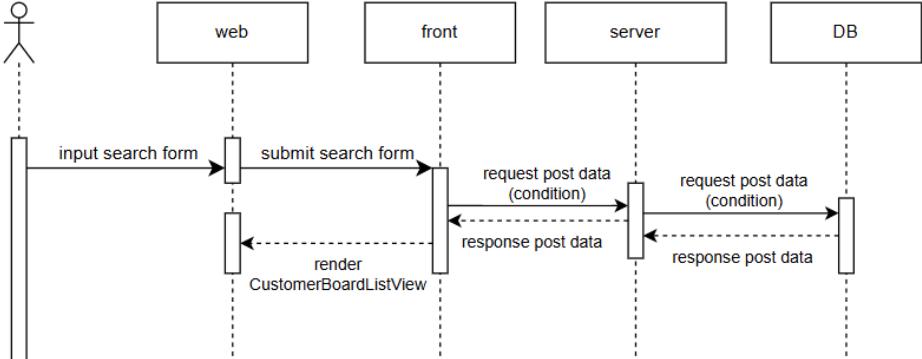
이 장은 sequence diagrams(이하 SD)와 그 설명을 제공한다. 고려사항은 다음과 같다.

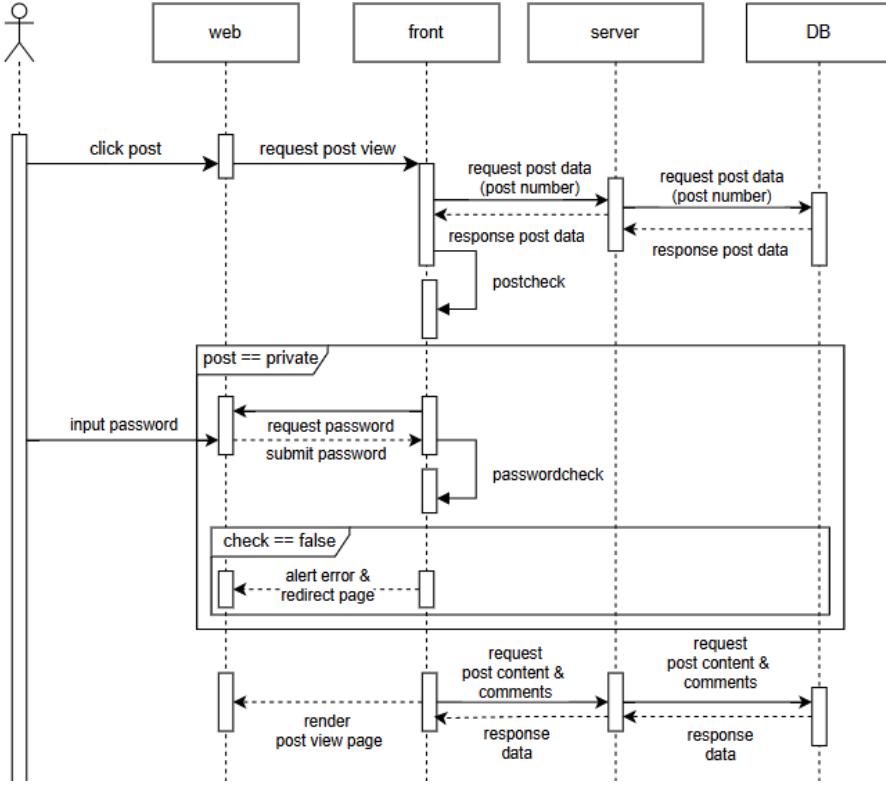
- 모든 SD는 user, 웹, 서버, DB로 표현된다.
- 즉, user와 웹 사이트 간의 통신, 웹 사이트와 서버간의 통신, 서버와 DB간의 통신을 나타낸다.
- 각 객체마다 보내는 요청의 순서를 중점적으로 설계하였으며 실제 각 객체 안에서 처리되는 상세 동작은 기술하지 않았다.
- 이를 통해 전체적인 프로젝트 구조, 흐름에 대해 알 수 있다.
- 모든 SD는 use case의 main/extension success scenario의 흐름을 따른다.

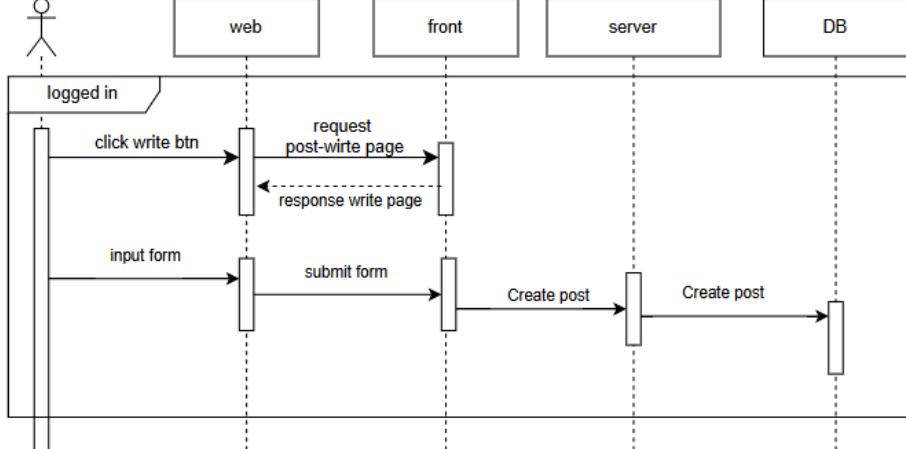
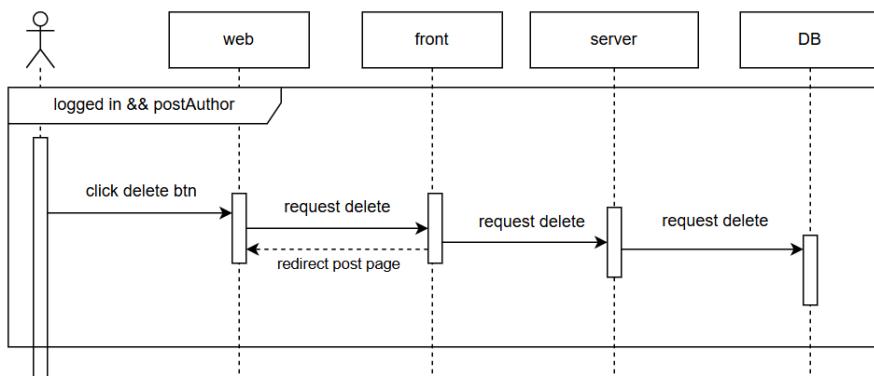
다음은 SD의 목록이다.

1. main SD
2. support
  - a. post list SD
  - b. post view SD
  - c. delete post SD
3. membership SD
4. login SD
5. classification SD
6. Bookinfo & detail SD
7. Logout SD
8. FindID SD
9. FindPwd SD
10. SetPwd SD
11. Mypage
  - a. update profile SD
  - b. login history SD
  - c. purchase history SD
12. Subscribe SD
13. Cart SD
14. Purchase SD
15. Group
  - a. group list SD
  - b. group info SD
  - c. group creation SD
  - d. group join SD
  - e. group manage SD
16. Plan
  - a. plan create SD
  - b. write plan content SD
17. FileViewer SD

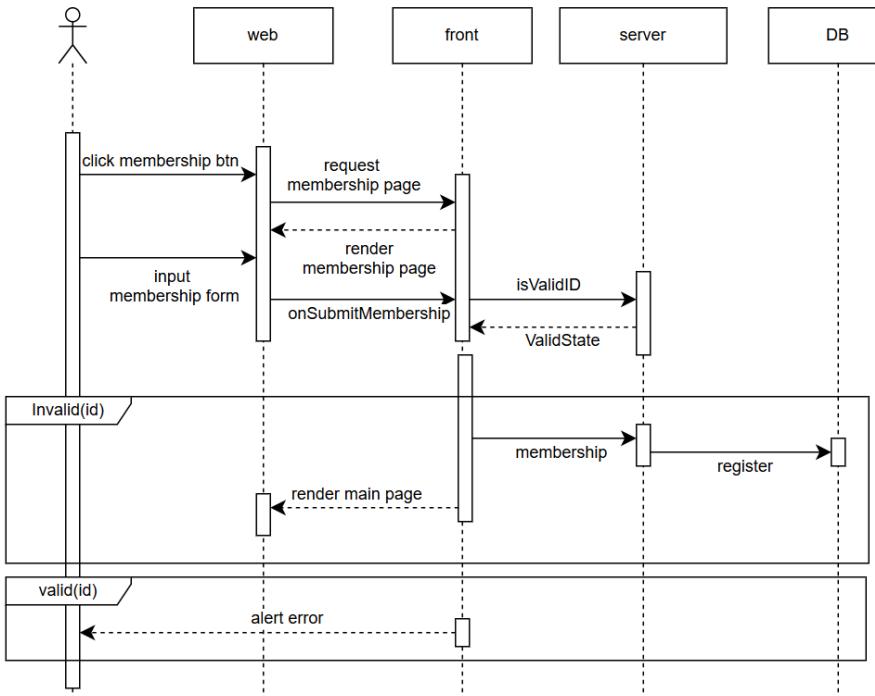
| main        |   |
|-------------|---|
| diagram     |   |
|             |  <pre> sequenceDiagram     participant User     participant Web     participant Front     participant Server     participant DB      User-&gt;&gt;Web: access main page     Web-&gt;&gt;Front: request main page     Note over Front: userid!=null     Front-&gt;&gt;Server: IsLoggedIn     Server--&gt;Front: userid     Front-&gt;&gt;Server: check login state     activate Front     if userid!=null then         Front-&gt;&gt;DB: getUserImage         DB--&gt;Front: Image         Front--&gt;User: Image     else         Front--&gt;User: render main page     end   </pre> |
| usecase     | 1   |
| description | <p>사용자가 시스템에 접속하는 경우를 나타내는 SD이다.</p> <p>시스템이 시작되면서 기능을 시작한다. 우선 로그인 되어있는지 확인하기 위해 세션에 저장된 정보를 가져온다. 계정이 존재하는 경우 DB에서 프로필 이미지를 가져온다. 계정이 존재하는 경우, 하지 않는 경우 모두 메인 페이지를 렌더링한다.</p>   |

| support |   |
|---------|---|
| 1       | support post list   |
| diagram |   |
|         |  <pre> sequenceDiagram     participant User     participant Web     participant Front     participant Server     participant DB      User-&gt;&gt;Web: input search form     Web-&gt;&gt;Front: submit search form     Front-&gt;&gt;Server: request post data (condition)     Server--&gt;Front: response post data     Front--&gt;User: response post data     Front--&gt;User: render CustomerBoardListView   </pre> |
| usecase | 11  |

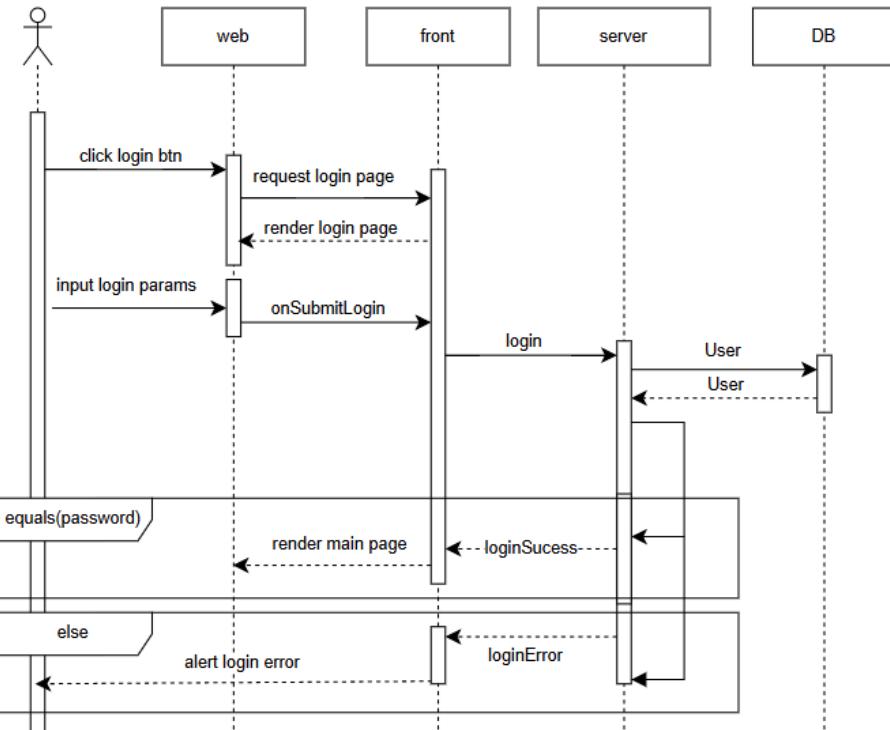
|             |   |
|-------------|---|
| description | 사용자가 고객 게시판 글 목록을 보기 위해 사용된다.<br>서버로 부터 검색 키워드에 맞는 리스트를 불러온다.<br>성공적으로 받으면 화면에 띄울 수 있게 한다.  |
| 2           | support post view   |
| diagram     |  <pre> sequenceDiagram     participant User     participant web     participant front     participant server     participant DB     User-&gt;&gt;web: click post     activate web     web-&gt;&gt;front: request post view     activate front     front-&gt;&gt;server: request post data (post number)     activate server     server-&gt;&gt;DB: request post data (post number)     activate DB     DB--&gt;&gt;front: response post data     activate front     front--&gt;&gt;web: response post data     deactivate web     deactivate front     deactivate server     deactivate DB     web--&gt;&gt;User: postcheck     activate User     if post == private then         User-&gt;&gt;front: input password         activate front         front-&gt;&gt;server: request password         activate server         server-&gt;&gt;front: submit password         activate front         front--&gt;&gt;User: passwordcheck         deactivate User         deactivate front         deactivate server         if check == false then             User--&gt;&gt;User: alert error &amp; redirect page         end     end     User--&gt;&gt;web: render post view page     activate web     web-&gt;&gt;front: request post content &amp; comments     activate front     front--&gt;&gt;server: response data     activate server     server--&gt;&gt;front: response data     activate front     front--&gt;&gt;web: response data     deactivate web     deactivate front     deactivate server   </pre> |
| usecase     | 11  |
| description | 사용자가 글 목록에서 특정 글을 선택하는 경우 시작된다. 서버로 부터 해당 글 정보를 불러오고 비밀글 여부를 체크한다.<br>비밀글이라면 암호를 사용자로부터 입력받는다. 암호가 일치하지 않는다면 extension scenario에 근거하여 에러 메시지를 내보낸다.<br>웹 화면에 글 정보를 보내어서 사용자가 글을 열람할 수 있게 한다.  |
| 3           | write post  |

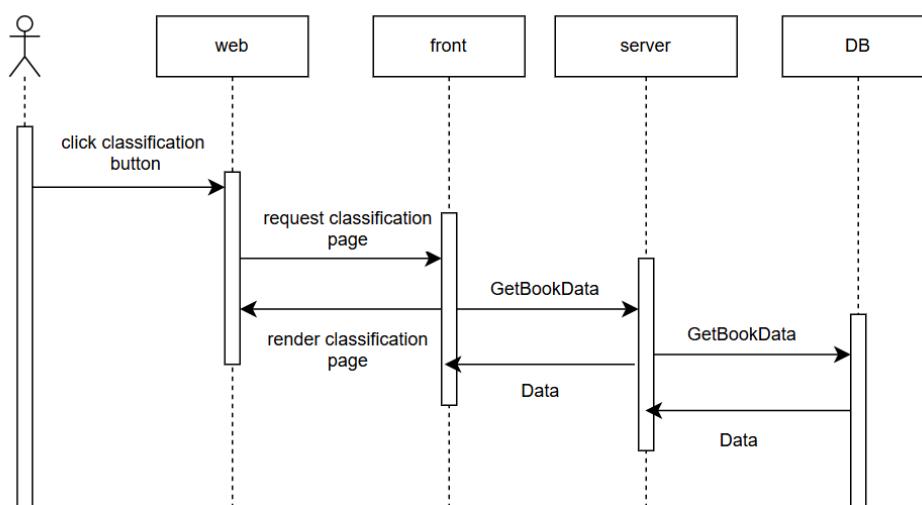
|             |  |
|-------------|--|
| diagram     |  <pre> sequenceDiagram     actor User     participant Web     participant Front     participant Server     participant DB     User-&gt;&gt;Web: click write btn     activate Web     Web-&gt;&gt;Front: request post-write page     deactivate Web     Front--&gt;&gt;User: response write page     User-&gt;&gt;Front: input form     activate Front     Front-&gt;&gt;Server: submit form     deactivate Front     activate Server     Server-&gt;&gt;Front: Create post     deactivate Server     Front--&gt;&gt;User: Create post   </pre> |
| usecase     | 11   |
| description | <p>사용자가 글 목록 페이지에서 글 작성 탭을 누르면 시작된다.<br/>   post condition에 따라 사용자가 로그인 한 상태여야 글 작성 버튼이 표시된다. 사용자가 글 생성 양식을 작성하고 제출 버튼을 누르면서 서버로 전송되어 DB에 저장된다.</p>  |
| 3           | delete post  |
| diagram     |  <pre> sequenceDiagram     actor User     participant Web     participant Front     participant Server     participant DB     User-&gt;&gt;Web: click delete btn     activate Web     Web-&gt;&gt;Front: request delete     deactivate Web     Front-&gt;&gt;Server: request delete     deactivate Front     Server-&gt;&gt;DB: request delete     deactivate Server   </pre>  |
| usecase     | 11   |
| description | <p>사용자는 글 목록 페이지나 특정 글 페이지에서 글 또는 댓글을 삭제할 수 있다. 사용자가 삭제 버튼을 누르는 경우 서버가 DB로 전송한다</p>  |

## membership

|             |  |
|-------------|--|
| diagram     |  <pre> sequenceDiagram     actor User     participant Web     participant Front     participant Server     participant DB      User-&gt;&gt;Web: click membership btn     activate Web     Web-&gt;&gt;Front: request membership page     activate Front     Front-&gt;&gt;Server: onSubmitMembership     activate Server     Server-&gt;&gt;DB: isValidID     activate DB     DB--&gt;&gt;Server: ValidState     deactivate DB     deactivate Server     Server--&gt;&gt;Front: membership     deactivate Front     Front--&gt;&gt;User: register     activate User     User--&gt;&gt;Front: Invalid(id)     activate Front     Front--&gt;&gt;User: render main page     deactivate User     User--&gt;&gt;Front: Valid(id)     activate Front     Front--&gt;&gt;User: alert error     deactivate User   </pre> |
| usecase     | 3  |
| description | <p>사용자가 회원가입을 할 때 sequence diagram이다. use case의 main success scenario에 근거하여 사용자가 회원가입을 위해 배너의 버튼을 클릭하면 시작된다.</p> <p>사용자가 회원가입 양식을 작성하면 웹은 서버로 아이디 중복 체크 요청을 보낸다.</p> <p>아이디가 유효하지 않다면 서버에 회원가입을 요청한다.</p> <p>use case의 extension scenario에 근거하여 아이디가 유효하다면 회원가입에 실패한다.</p>   |

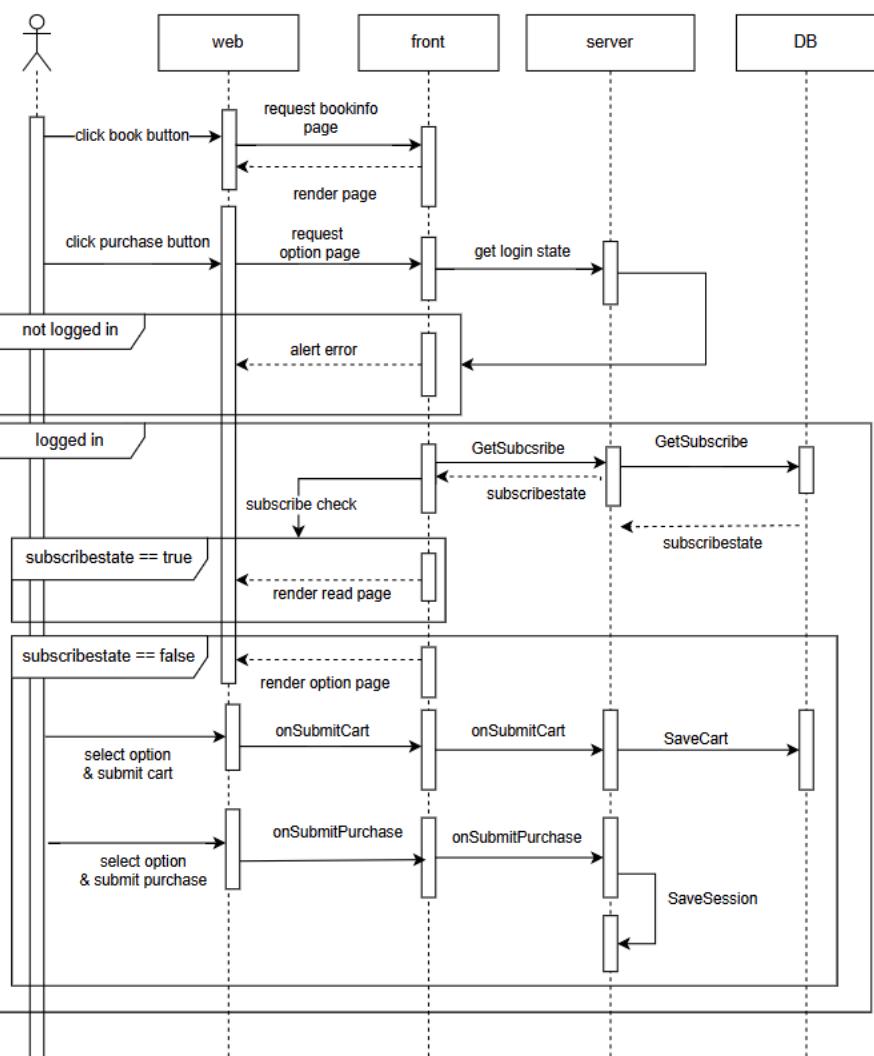
**login**

|             |   |
|-------------|---|
| diagram     |  <pre> sequenceDiagram     participant User     participant web     participant front     participant server     participant DB     User-&gt;&gt;web: click login btn     activate web     web-&gt;&gt;front: request login page     activate front     front-&gt;&gt;User: render login page     deactivate front     User-&gt;&gt;front: input login params     activate front     front-&gt;&gt;server: onSubmitLogin     activate server     server-&gt;&gt;User: login     activate User     User-&gt;&gt;DB: User     activate DB     DB--&gt;&gt;User: User     deactivate DB     User--&gt;&gt;front: equals(password)     activate front     User--&gt;&gt;front: render main page     deactivate front     User--&gt;&gt;User: loginSuccess     deactivate User     User--&gt;&gt;User: else     User--&gt;&gt;User: alert login error     deactivate User     </pre> |
| usecase     | 4   |
| description | <p>사용자가 시스템에 로그인 하려할 때 동작하는 SD이다. 사용자가 로그인 버튼을 누른 뒤 로그인 페이지에서 아이디, 비밀번호를 입력하면 서버는 유효한 사용자인지 DB를 통해 확인한다. 사용자가 유효하다면 메인 페이지를 띄우고 아니라면 로그인 에러를 띄운다.</p>   |

| classification |   |
|----------------|---|
| diagram        |  <pre> sequenceDiagram     participant User     participant web     participant front     participant server     participant DB     User-&gt;&gt;web: click classification button     activate web     web-&gt;&gt;front: request classification page     activate front     front-&gt;&gt;User: render classification page     deactivate front     User--&gt;&gt;server: GetBookData     activate server     server--&gt;&gt;User: Data     User--&gt;&gt;DB: GetBookData     activate DB     DB--&gt;&gt;User: Data     deactivate DB     </pre> |
| usecase        | 14  |

|             |  |
|-------------|--|
| description | <p>사용자가 카테고리별 분류된 도서 목록을 열람하는 경우 동작하는 SD이다. 사용자가 메인 페이지에서 카테고리별 버튼을 선택하는 경우 시작된다.</p> <p>서버는 DB로 데이터를 불러오고 웹에 전달해 사용자가 추천받을 수 있게 한다.</p> |
|-------------|--|

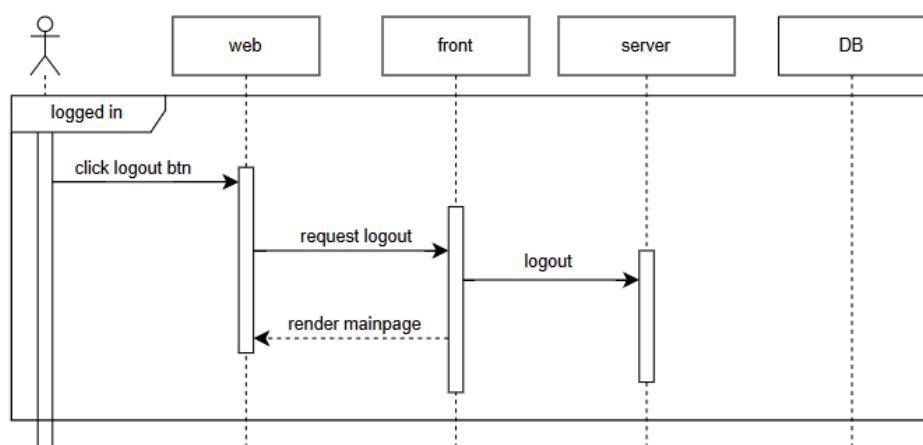
### BookInfo & detail

|             |  |
|-------------|--|
| diagram     |  <pre> sequenceDiagram     actor User     participant Web as web     participant Front as front     participant Server as server     participant DB as DB      User-&gt;&gt;Web: click book button     Web-&gt;&gt;Front: request bookinfo page     Front-&gt;&gt;User: render page     User-&gt;&gt;Web: click purchase button     Web-&gt;&gt;Front: request option page     Front-&gt;&gt;User: get login state     Note over User: not logged in     User--&gt;&gt;User: alert error      User--&gt;&gt;Web: logged in     Web-&gt;&gt;Front: GetSubscribe     Front-&gt;&gt;User: subscribe check     User--&gt;&gt;User: subscribestate == true     User--&gt;&gt;Web: subscribestate == false     User--&gt;&gt;Front: render read page     Front-&gt;&gt;User: render option page     User-&gt;&gt;Web: select option &amp; submit cart     Web-&gt;&gt;Front: onSubmitCart     Front-&gt;&gt;User: onSubmitCart     User-&gt;&gt;Web: onSubmitPurchase     Web-&gt;&gt;Front: onSubmitPurchase     Front-&gt;&gt;User: SaveCart     User--&gt;&gt;User: SaveSession     </pre> |
| usecase     | 15, 16   |
| description | <p>사용자가 특정 책 페이지에 접근하는 경우 나타내는 SD이다. use case #15의 main success scenario에 근거하여 특정 책을 선택하는 경우 시작된다.</p> <p>서버는 DB로 부터 책 정보에 관한 내용을 받아오고 페이지를 렌더링한다.</p> <p>use case #16에 근거하여 구매옵션을 선택하는 버튼을 클릭하는</p>   |

경우 구독 여부를 체크한다. 서버는 DB로 부터 구독 정보를 가져온다.  
 구독중인 사용자인 경우 책을 바로 읽을 수 있는 페이지를 렌더링한다.  
 아닌 경우 구매 옵션 페이지를 렌더링한다.  
 사용자가 구매 옵션을 선택하고 장바구니를 선택하는 경우 서버에 장바구니 정보를 전송하고 DB에 저장한다.  
 사용자가 구매 옵션을 선택하고 구매하기를 선택하는 경우 서버에 구매 정보를 전송하고 서버는 세션에 저장한다.

## Logout

### diagram



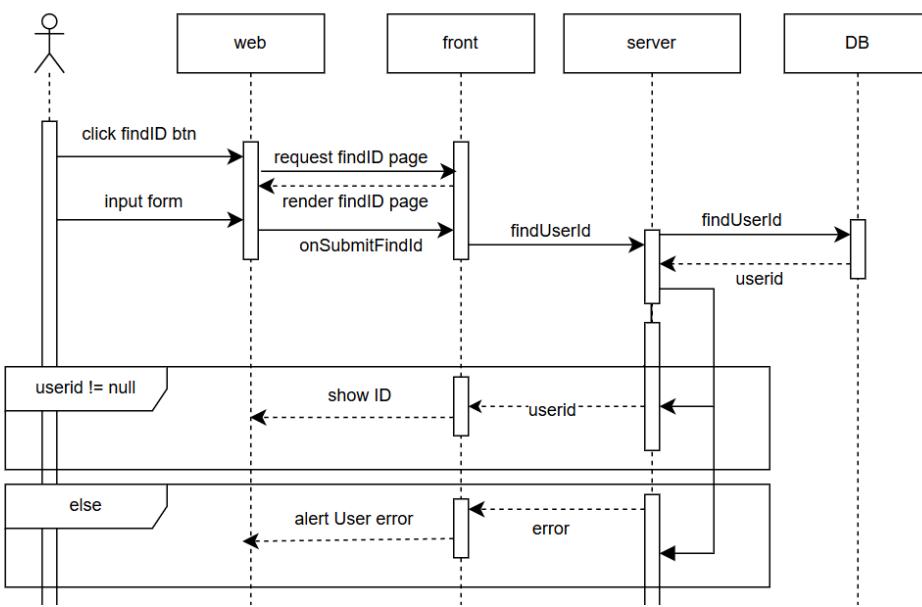
### usecase

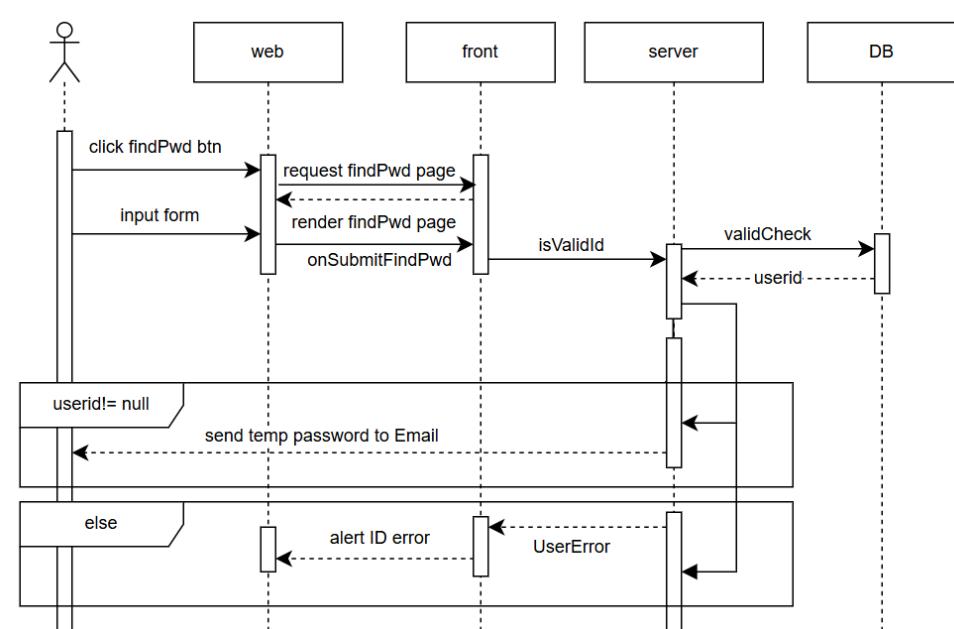
13

### description

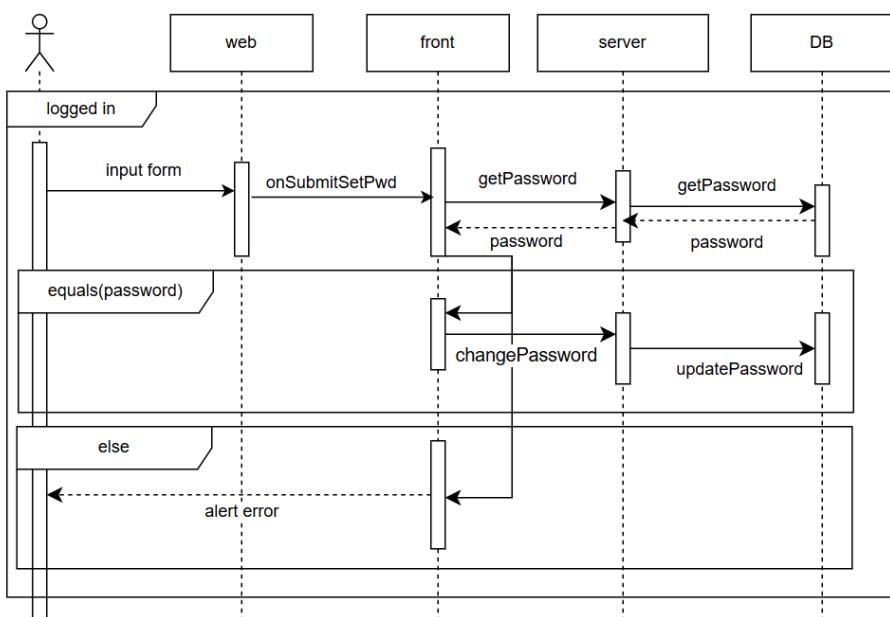
사용자가 로그아웃을 하려할 때 동작하는 SD 이다. post condition에 따라 로그인된 상태여야 한다. 사용자로 부터 로그아웃을 요청하면 서버에서 현재 로그인 상태를 해제한다.

## Find ID

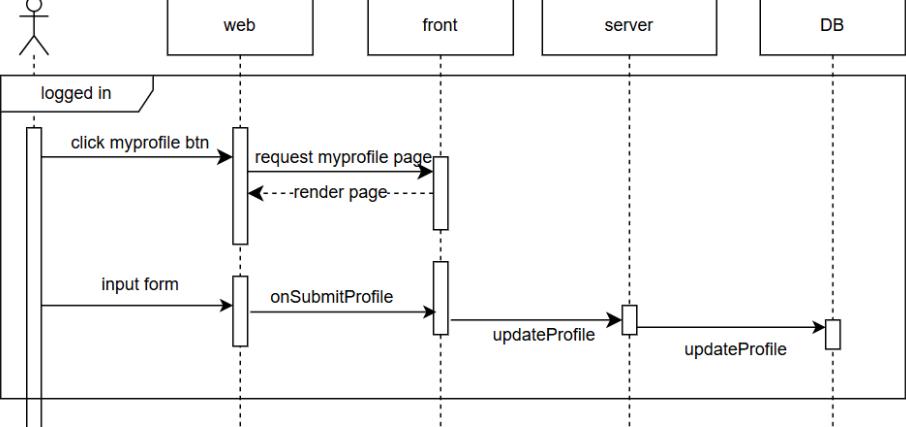
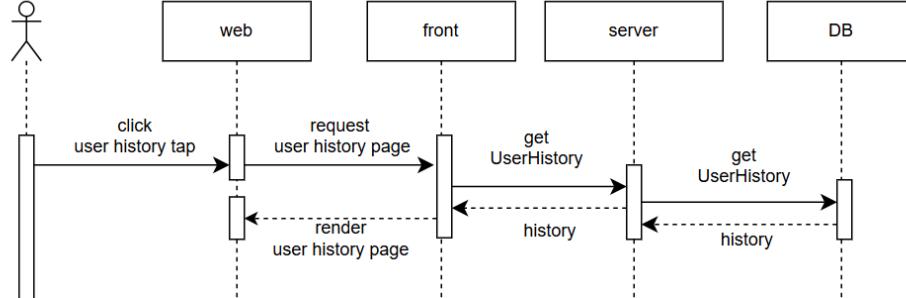
|             |   |
|-------------|---|
| diagram     |  <pre> sequenceDiagram     participant User     participant web     participant front     participant server     participant DB     User-&gt;&gt;web: click findID btn     activate web     web-&gt;&gt;front: request findID page     activate front     front-&gt;&gt;User: render findID page     User-&gt;&gt;front: input form     activate front     front-&gt;&gt;server: onSubmitFindId     activate server     server-&gt;&gt;DB: findUserId     activate DB     DB-&gt;&gt;server: userid     deactivate DB     server--&gt;&gt;front: userid     deactivate server     front--&gt;&gt;User: show ID     deactivate front     User--&gt;&gt;server: userid != null     activate server     server--&gt;&gt;User: error     deactivate server     User--&gt;&gt;front: alert User error     deactivate User   </pre> |
| usecase     |   |
| description | <p>사용자가 아이디를 찾을 때 동작하는 SD이다.</p> <p>사용자가 아이디를 제외한 회원정보를 입력하면 서버로 부터 정보가 일치하는 유저를 반환한다.</p>  |

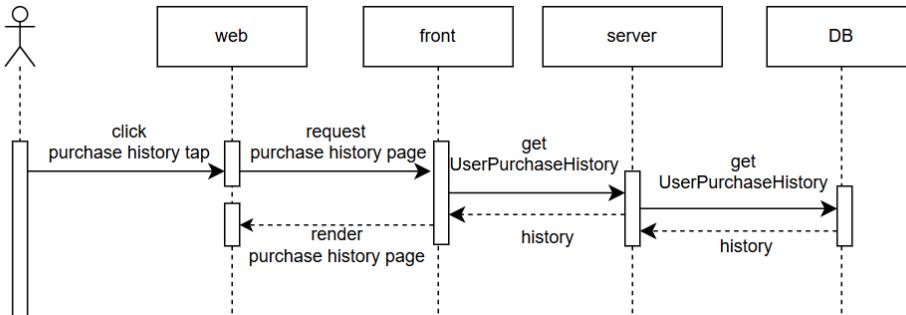
| <b>Find PWD</b> |   |
|-----------------|---|
| diagram         |  <pre> sequenceDiagram     participant User     participant web     participant front     participant server     participant DB     User-&gt;&gt;web: click findPwd btn     activate web     web-&gt;&gt;front: request findPwd page     activate front     front-&gt;&gt;User: render findPwd page     User-&gt;&gt;front: input form     activate front     front-&gt;&gt;server: onSubmitFindPwd     activate server     server-&gt;&gt;DB: isValidId     activate DB     DB-&gt;&gt;server: userid     deactivate DB     server--&gt;&gt;front: validCheck     deactivate server     front--&gt;&gt;User: userid!= null     deactivate front     User--&gt;&gt;server: send temp password to Email     deactivate User     User--&gt;&gt;front: alert ID error     deactivate User   </pre> |
| usecase         |   |
| description     | <p>사용자가 비밀번호를 찾으려 할 때 동작하는 SD이다.</p> <p>use case 의 main success scenario에 근거하여 사용자가</p>   |

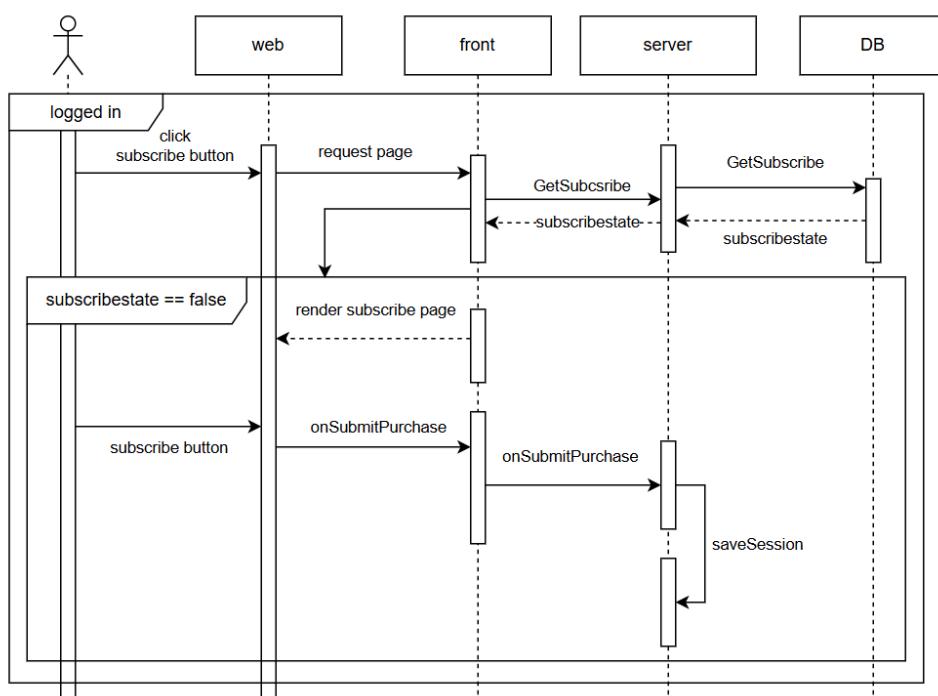
|  |  |
|--|--|
|  | <p>회원정보를 입력하면 서버는 DB로 부터 유효한 사용자를 찾는다.<br/>     success scenario에 근거하여 유효한 사용자면 사용자의 이메일로<br/>     임시 비밀번호를 보낸다.<br/>     extension scenario에 근거하여 유효하지 않은 사용자라면 에러를<br/>     보낸다.</p> |
|--|--|

| <b>Set PWD</b> |  |
|----------------|--|
| diagram        |  <pre> sequenceDiagram     participant User     participant web     participant front     participant server     participant DB      User-&gt;&gt;web: logged in     activate web     User-&gt;&gt;front: input form     activate front     front-&gt;&gt;server: onSubmitSetPwd     activate server     server-&gt;&gt;DB: getPassword     activate DB     DB--&gt;&gt;server: password     deactivate DB     server--&gt;&gt;front: equals(password)     deactivate server     front--&gt;&gt;server: changePassword     activate server     server--&gt;&gt;DB: updatePassword     deactivate server     DB--&gt;&gt;front: alert error     deactivate DB     </pre> |
| usecase        | 7  |
| description    | <p>사용자가 비밀번호를 재설정할 때 동작하는 SD이다. main success scenario에 근거하여 사용자가 마이페이지에서 비밀번호 변경을 시도할 때 시작된다. 사용자가 비밀번호 변경 정보(현재 비밀번호, 변경할 비밀번호, 재 확인)을 입력하면 서버로 부터 확인한 현재 비밀번호와 비교한다.<br/>     동일하다면 비밀번호를 변경하고 일치하지 않다면 extension scenario에 근거하여 에러 메시지를 보낸다.</p>  |

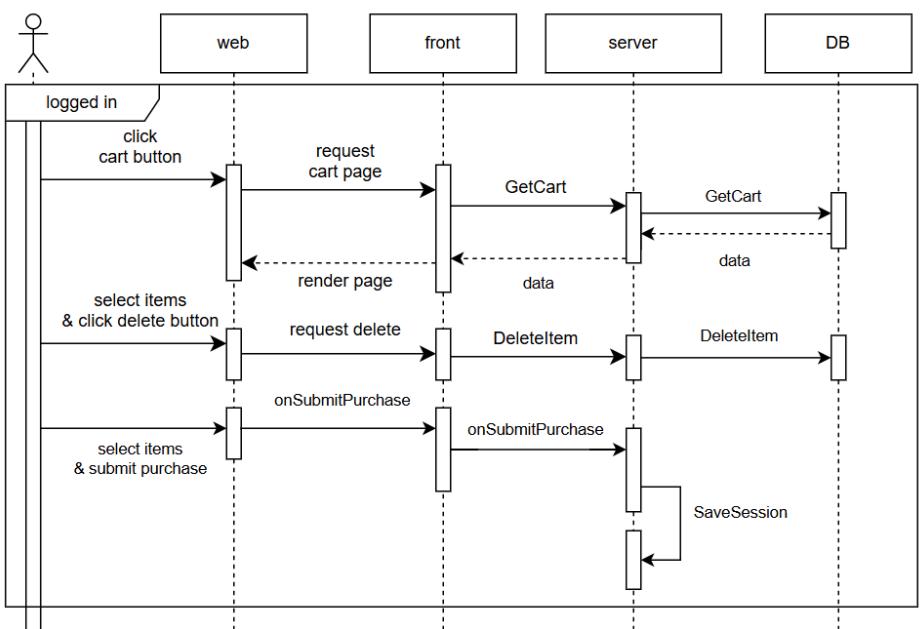
| <b>My page</b> |                |
|----------------|----------------|
| 1              | update profile |

|             |  |
|-------------|--|
| diagram     |  <pre> sequenceDiagram     participant User     participant Web     participant Front     participant Server     participant DB      User-&gt;&gt;Web: logged in     activate Web     User-&gt;&gt;Front: click myprofile btn     activate Front     Front-&gt;&gt;Server: request myprofile page     activate Server     Note over Server: &lt;--render page--&gt;     Server-&gt;&gt;Front: updateProfile     deactivate Front     deactivate Server     User-&gt;&gt;Front: input form     activate Front     Front-&gt;&gt;Server: onSubmitProfile     activate Server     Note over Server: &lt;--updateProfile--&gt;     Server-&gt;&gt;Front: updateProfile     deactivate Front     deactivate Server   </pre> |
| usecase     | 1,4,2  |
| description | <p>사용자가 자신의 프로필 페이지에 접근하는 경우 나타내는 SD이다.</p> <p>user class description에서 &lt;Use case #1&gt;와 &lt;Use case #4&gt;와 &lt;Use case #2&gt;이다.</p> <p>case 1에 근거하여 사용자는 로그인을 통해 프로필 페이지에 접근할 수 있다.</p> <p>case 4,5에 근거하여 사용자는 프로필 이미지, 닉네임을 수정할 수 있다.</p>  |
| 2           | login history  |
| diagram     |  <pre> sequenceDiagram     participant User     participant Web     participant Front     participant Server     participant DB      User-&gt;&gt;Web: click user history tap     activate Web     Web-&gt;&gt;Front: request user history page     activate Front     Note over Front: &lt;--render user history page--&gt;     Front-&gt;&gt;Server: get UserHistory     activate Server     Note over Server: &lt;--history--&gt;     Server-&gt;&gt;Front: get UserHistory     deactivate Front     deactivate Server   </pre>   |
| usecase     | 8  |
| description | <p>사용자가 자신의 활동 기록에 접근하는 경우 나타내는 SD이다.</p> <p>main success scenario에 근거하여 사용자가 마이페이지에서 활동 내역을 열람할 때 시작된다. 서버는 DB로부터 사용자의 활동기록을 가져오고 web에 전달한다.</p>  |
| 3           | purchase history   |

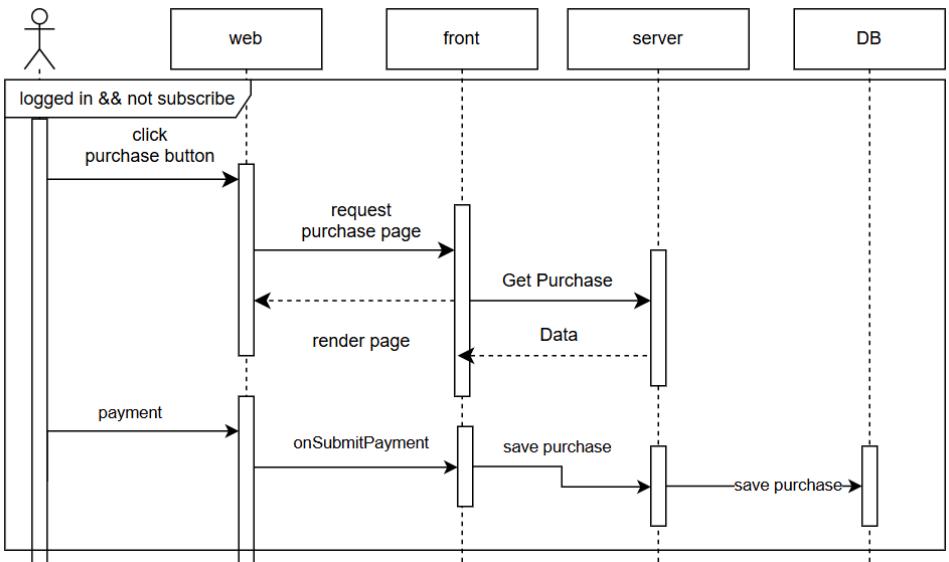
|             |   |
|-------------|---|
| diagram     |  <pre> sequenceDiagram     participant User     participant web     participant front     participant server     participant DB     User-&gt;&gt;web: click purchase history tap     activate web     web-&gt;&gt;front: request purchase history page     activate front     front-&gt;&gt;server: get UserPurchaseHistory     activate server     server-&gt;&gt;DB: get UserPurchaseHistory     activate DB     DB--&gt;&gt;server: history     deactivate DB     server--&gt;&gt;front: history     deactivate server     front--&gt;&gt;User: render purchase history page     deactivate front   </pre> |
| usecase     | 9   |
| description | <p>사용자가 자신의 구매/대여 내역에 접근하는 경우 나타내는 SD이다. main success scenario에 근거하여 사용자가 마이페이지에서 구매/대여 내역을 열람할 때 시작된다. 서버는 DB로부터 사용자의 구매/대여 내역을 가져오고 web에 전달한다.</p>  |

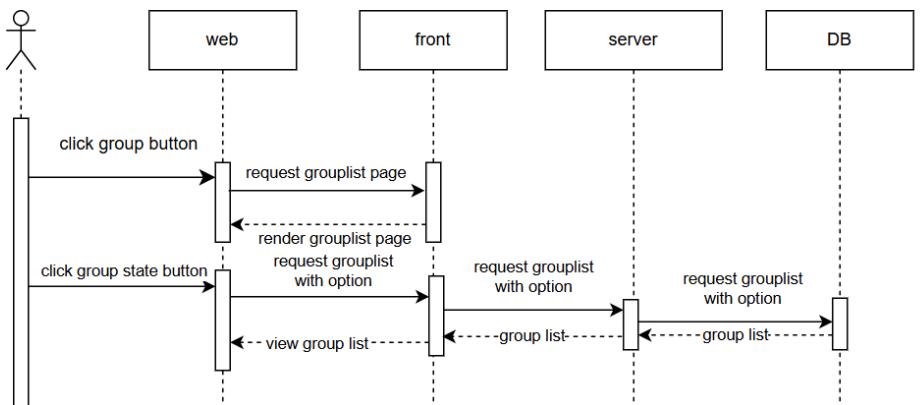
| <b>Subscribe</b> |   |
|------------------|---|
| diagram          |  <pre> sequenceDiagram     participant User     participant web     participant front     participant server     participant DB     User-&gt;&gt;web: logged in     User-&gt;&gt;web: click subscribe button     activate web     web-&gt;&gt;front: request page     activate front     front-&gt;&gt;server: GetSubscribe     activate server     server-&gt;&gt;DB: GetSubscribe     activate DB     DB--&gt;&gt;server: subscribestate     deactivate DB     server--&gt;&gt;front: subscribestate     deactivate server     front--&gt;&gt;User: render subscribe page     deactivate front     User-&gt;&gt;web: subscribe button     activate web     web-&gt;&gt;front: onSubmitPurchase     activate front     front-&gt;&gt;server: onSubmitPurchase     activate server     server-&gt;&gt;DB: saveSession     activate DB     DB--&gt;&gt;server:      deactivate DB     server--&gt;&gt;front:      deactivate server   </pre> |
| usecase          | 17  |
| description      | <p>사용자가 이용권을 구매하는 경우를 나타내는 SD이다. &lt;use case #17&gt;에 해당한다. main success에 근거하여 사용자가 메인 배너의 이용권 구매하기 버튼을 누른 경우 시작된다. 서버는 DB로부터 현재 사용자가 구독중인 사용자인지 일반 사용자인지 확인한다. 일반 사용자인 경우에만 이용권 페이지를 렌더링한다.</p>   |

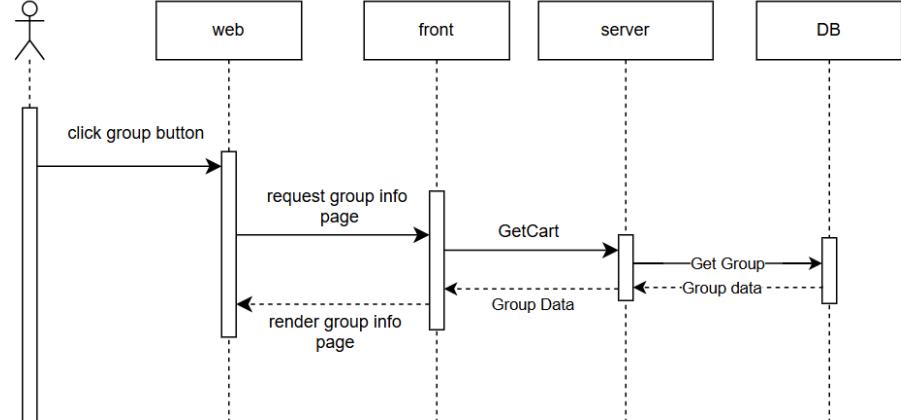
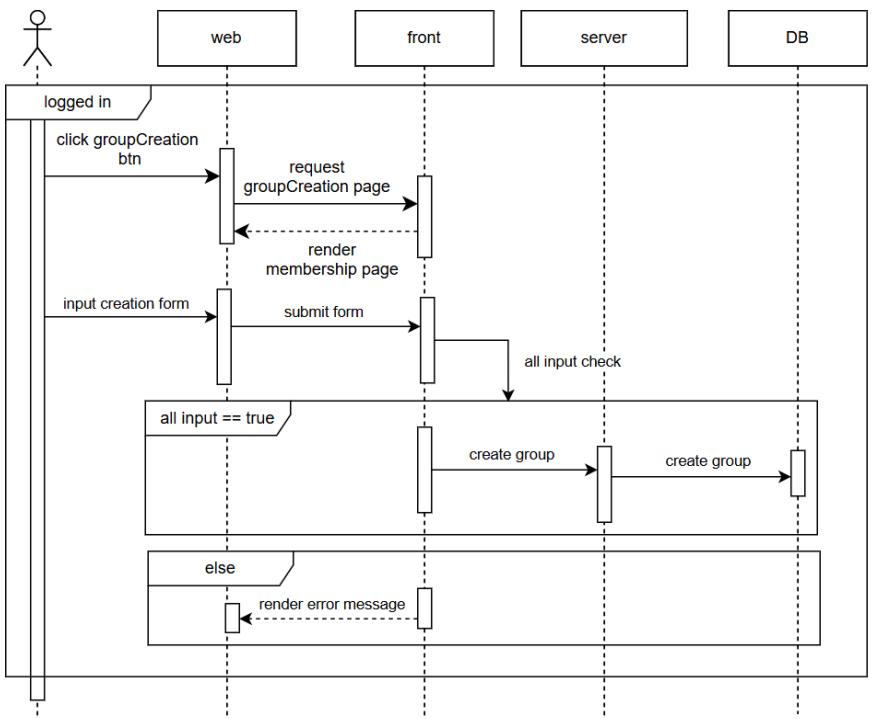
|  |   |
|--|---|
|  | 사용자가 구독 버튼을 누르는 경우 서버는 이용권 구매 정보를 세션에 저장한다. |
|--|---|

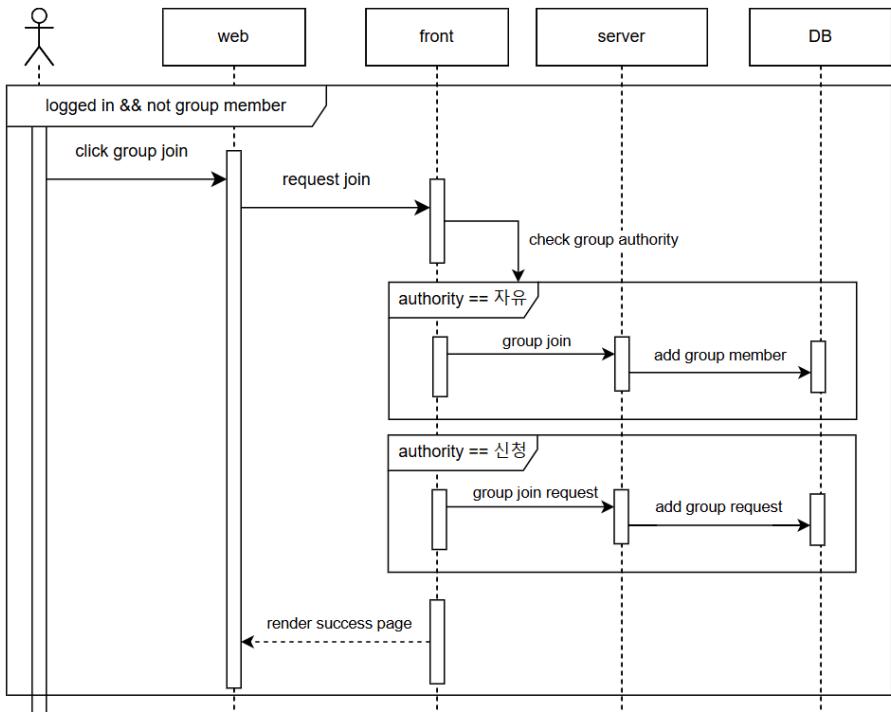
| Cart        |  |
|-------------|--|
| diagram     |  |
|             |  <pre> sequenceDiagram     participant User     participant Web     participant Front     participant Server     participant DB      User-&gt;&gt;Web: logged in     activate Web     User-&gt;&gt;Front: click cart button     activate Front     Web-&gt;&gt;Front: request cart page     activate Front     Front-&gt;&gt;Server: GetCart     activate Server     Server-&gt;&gt;Front: data     deactivate Front     Front-&gt;&gt;User: render page     deactivate User     User-&gt;&gt;Front: select items &amp; click delete button     activate Front     User-&gt;&gt;Server: request delete     activate Server     Server-&gt;&gt;Front: DeleteItem     activate Front     Front-&gt;&gt;User: DeleteItem     deactivate User     User-&gt;&gt;Front: onSubmitPurchase     activate Front     User-&gt;&gt;Server: onSubmittedPurchase     activate Server     Server-&gt;&gt;Front: onSubmittedPurchase     deactivate Front     User-&gt;&gt;DB: SaveSession     deactivate DB   </pre> |
| usecase     | 18   |
| description | 사용자가 장바구니에 접근하는 경우를 나타내는 SD이다. main success scenario에 근거하여 사용자가 메인 배너의 장바구니 버튼을 클릭하는 경우 시작된다. 서버는 DB에 저장된 장바구니 정보를 가져오고 화면에 내보낸다. 사용자는 장바구니에 저장된 항목을 삭제할 수 있다.   |

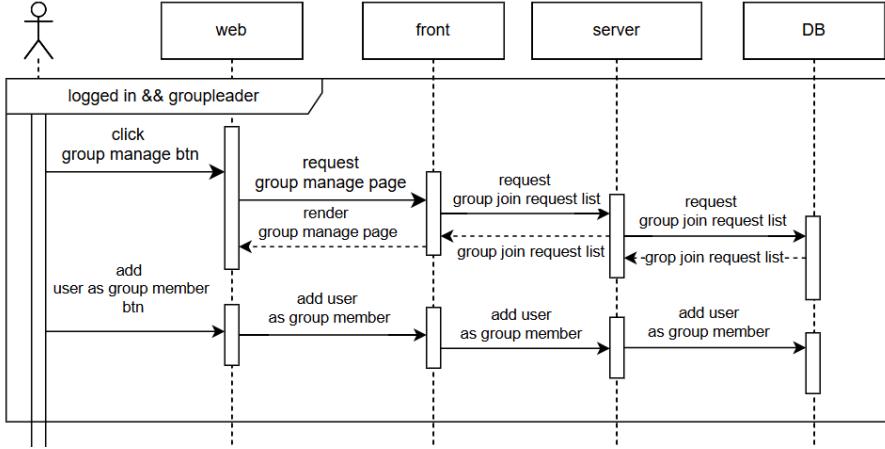
## Purchase

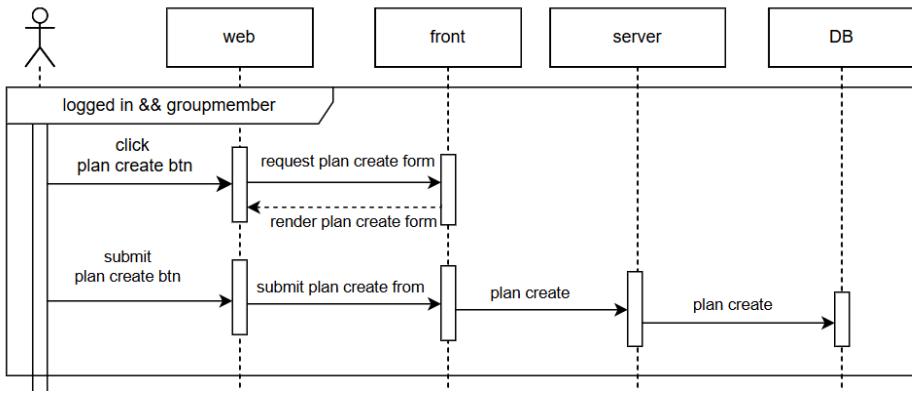
|             |   |
|-------------|---|
| diagram     |  <pre> sequenceDiagram     participant User     participant web     participant front     participant server     participant DB     User-&gt;&gt;web: logged in &amp;&amp; not subscribe     activate web     web-&gt;&gt;front: click purchase button     activate front     front-&gt;&gt;server: request purchase page     activate server     server-&gt;&gt;front: Get Purchase     activate front     front-&gt;&gt;User: render page     deactivate User     User-&gt;&gt;server: payment     activate server     server-&gt;&gt;front: onSubmitPayment     activate front     front-&gt;&gt;server: save purchase     activate server     server--&gt;&gt;DB: save purchase     deactivate front     deactivate server   </pre> |
| usecase     | 19  |
| description | <p>사용자가 구매를 하려는 경우를 나타내는 SD이다. main success scenario에 근거하여 사용자가 구매하기 버튼을 누르는 경우 시작된다. 서버는 세션에 저장된 구매 정보를 불러오고 화면에 내보낸다. 사용자가 결제 버튼을 누르고 결제를 하며 서버에 구매정보가 저장된다.</p>  |

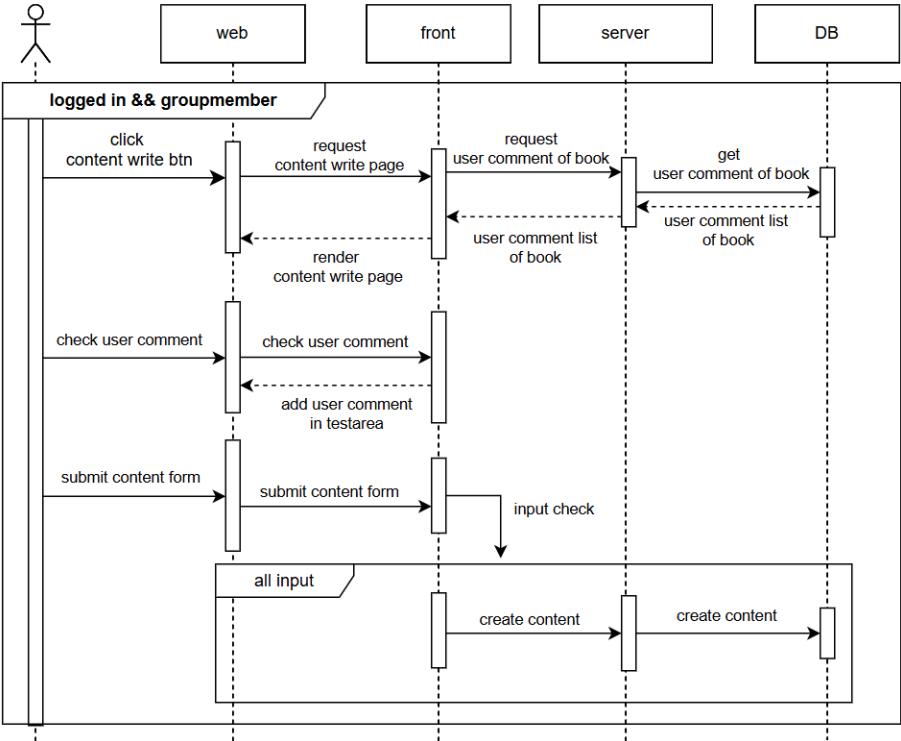
| Group       |  |
|-------------|--|
| 1           | group list   |
| diagram     |  <pre> sequenceDiagram     participant User     participant web     participant front     participant server     participant DB     User-&gt;&gt;web: click group button     activate web     web-&gt;&gt;front: request grouplist page     activate front     front-&gt;&gt;server: render grouplist page     activate server     server-&gt;&gt;front: request grouplist with option     activate front     front--&gt;&gt;User: view group list     deactivate User     User-&gt;&gt;server: click group state button     activate server     server-&gt;&gt;front: request grouplist with option     activate front     front--&gt;&gt;User: group list     deactivate User   </pre> |
| usecase     | 21   |
| description | <p>사용자가 그룹 목록을 열람하는 경우를 나타내는 SD이다. main success scenario에 근거하여 사용자가 메인 배너의 그룹 페이지로 이동하는 버튼을 누르면 시작된다. 서버는 DB에 저장된 그룹 목록을 불러오고 화면에 내보낸다.</p>  |

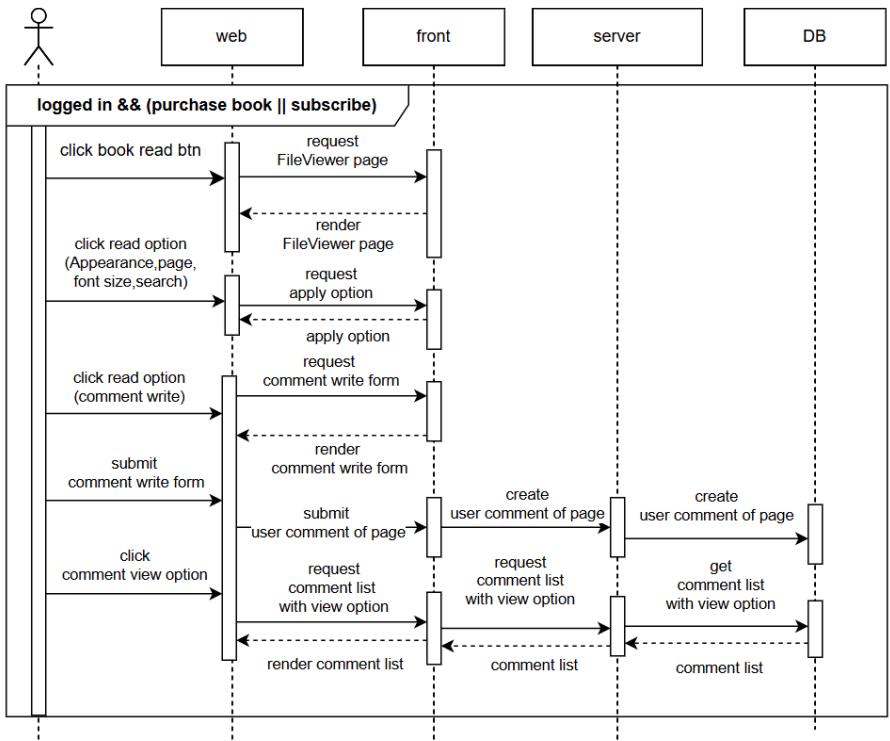
|             |  |
|-------------|--|
| 2           | group info   |
| diagram     |  <pre> sequenceDiagram     participant User     participant web     participant front     participant server     participant DB     User-&gt;&gt;web: click group button     activate web     web-&gt;&gt;front: request group info page     activate front     front-&gt;&gt;server: GetCart     activate server     server-&gt;&gt;DB: Get Group     activate DB     DB--&gt;&gt;server: Group data     deactivate DB     server--&gt;&gt;front: Group Data     deactivate server     front--&gt;&gt;User: render group info page     deactivate front   </pre>  |
| usecase     | 22   |
| description | 사용자가 그룹 소개를 확인하는 경우를 나타내는 SD이다. main success scenario에 근거하여 그룹 목록에서 그룹을 면다. 버는 DB에 된 그룹 정보를 가져오고 화면에 출력한다.   |
| 3           | group creation   |
| diagram     |  <pre> sequenceDiagram     participant User     participant web     participant front     participant server     participant DB     User-&gt;&gt;web: logged in     activate web     web-&gt;&gt;front: click groupCreation btn     activate front     front-&gt;&gt;server: request groupCreation page     activate server     server--&gt;&gt;front: render membership page     deactivate server     front-&gt;&gt;User: input creation form     activate User     User-&gt;&gt;front: submit form     activate front     front-&gt;&gt;server: all input check     activate server     if all input == true then         server-&gt;&gt;front: create group         activate front         front--&gt;&gt;server: create group         deactivate front     else         server--&gt;&gt;User: render error message     end     deactivate User   </pre> |
| usecase     | 23   |
| description | 사용자가 그룹을 생성하는 경우를 나타내는 SD이다. main success scenario에 근거하여 그룹 목록 페이지에서 그룹을 생성하는 버튼을  |

|             |   |
|-------------|---|
|             | <p>누르면 시작된다. 사용자는 그룹 이름, 책, 권한, 기한을 입력하고 제출버튼을 누른다. 만약 입력 양식 중 하나라도 입력이 안된 경우에러 팝업을 띄우고 다시 입력할 수 있게 한다.</p>   |
| 4           | group join  |
| diagram     |  <pre> sequenceDiagram     actor User     participant Web     participant Front     participant Server     participant DB      User-&gt;&gt;Web: logged in &amp;&amp; not group member     activate Web     User-&gt;&gt;Front: click group join     activate Front     Front-&gt;&gt;Server: request join     activate Server     Note over Server: check group authority     Note over Server: authority == 자유     Server-&gt;&gt;Front: group join     activate Front     Note over Server: authority == 신청     Server-&gt;&gt;Front: group join request     activate Front     Front-&gt;&gt;DB: add group member     activate DB     DB--&gt;&gt;Front: render success page     deactivate Front     deactivate Server     deactivate DB   </pre> <p>The sequence diagram illustrates the process for a user to join a group. It starts with the user logging in and checking if they are a group member. If not, the user clicks the 'group join' button on the web interface. This triggers a 'request join' message to the front-end. The front-end then checks the group authority. If the authority is '자유' (free), it performs a 'group join' operation directly on the server, which then adds the user as a group member. If the authority is '신청' (application), it sends a 'group join request' to the server, which adds the request to the group's pending applications. Finally, the front-end renders a success page.</p> |
| usecase     | 26  |
| description | <p>사용자가 그룹에 가입하는 경우를 나타내는 SD이다. 사용자가 그룹 참여하기 버튼을 누르면 시작된다. main success scenario에 근거하여 그룹 가입 조건이 자유이면 즉시 서버로 전송되어 그룹 멤버 테이블에 추가되고 가입 조건이 신청이면 그룹 가입 신청 테이블에 추가되어 리더가 승인할 때까지 기다린다.</p>  |
| 5           | group manage  |

|             |  |
|-------------|--|
| diagram     |    |
| usecase     | 24   |
| description | <p>사용자가 그룹을 관리할 때 사용하는 SD이다. precondition에 근거하여 로그인된 상태이며 해당 그룹의 관리자여야 한다. extension scenario에 근거하여 그룹의 관리자가 아닌 경우 관리하기 버튼을 노출시키지 않는다. main success scenario에 근거하여 사용자는 그룹 페이지에서 관리하기 버튼을 누른다. 서버는 해당 그룹에 신청 요청을 보낸 유저들의 목록을 가져온다. 사용자는 유저를 선택해 그룹에 가입시킬 수 있다.</p> |

| Plan        |   |
|-------------|---|
| 1           | PlanCreate  |
| diagram     |                                   |
| usecase     | 25  |
| description | <p>사용자가 해당 그룹에 대해 플랜을 생성하는 경우를 나타내는 SD이다. 사용자가 그룹 페이지에서 플랜 생성하기 버튼을 누르면 시작된다. precondition에 근거하여 사용자는 해당 그룹의 멤버여야</p> |

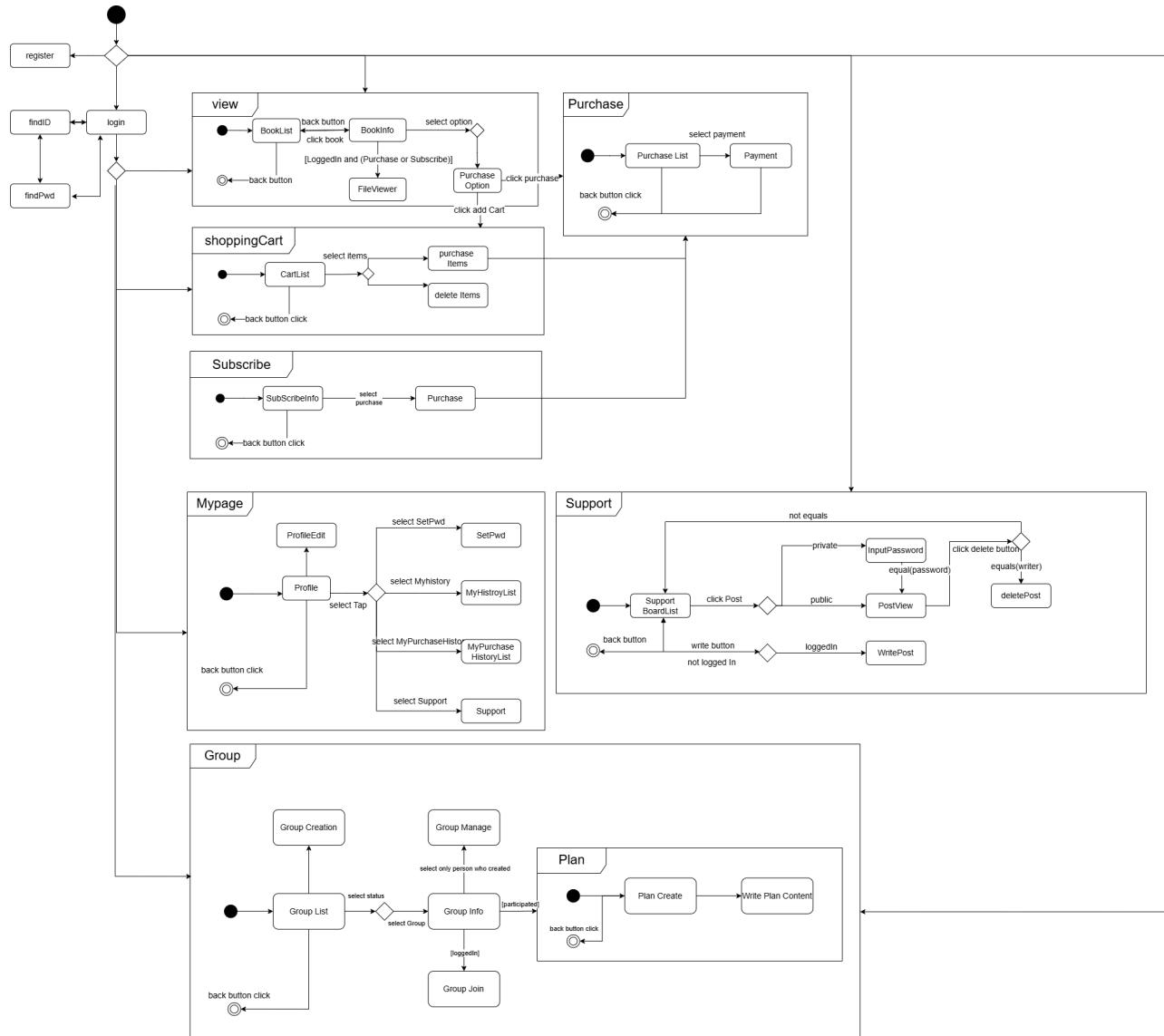
|             |  |
|-------------|--|
|             | <p>한다. 사용자는 플랜 제목을 입력하고 제출하기를 누른다. extension scenario에 근거하여 그룹의 멤버가 아닌 경우 생성하기 버튼을 노출시키지 않는다. main success scenario에 근거하여 플랜 제목이 빈 문자열이 아닌 경우 서버로 플랜이 전송된다.</p>  |
| 2           | write plan content   |
| diagram     |  <pre> sequenceDiagram     participant User     participant Web     participant Front     participant Server     participant DB      User-&gt;&gt;Web: logged in &amp;&amp; groupmember     activate Web     Web-&gt;&gt;Front: click content write btn     activate Front     Front-&gt;&gt;Server: request content write page     activate Server     Server-&gt;&gt;DB: get user comment of book     activate DB     DB--&gt;&gt;Server: user comment list of book     deactivate DB     Server--&gt;&gt;Front: user comment of book     deactivate Server     Front--&gt;&gt;User: render content write page     deactivate Front     User-&gt;&gt;Front: check user comment     activate Front     Front-&gt;&gt;Server: check user comment     activate Server     Server--&gt;&gt;Front: add user comment in testarea     deactivate Server     Front--&gt;&gt;User: submit content form     activate User     User-&gt;&gt;Front: submit content form     activate Front     Front--&gt;&gt;Server: submit content form     activate Server     Server--&gt;&gt;Front: input check     deactivate Server     User--&gt;&gt;Front: all input     activate Front     User--&gt;&gt;Server: create content     activate Server     Server--&gt;&gt;User: create content     deactivate Server     </pre> <p>The sequence diagram illustrates the process of writing a plan content. It starts with a user logging in and becoming a group member. The user then clicks a 'content write' button on the web interface, which triggers a 'request content write page' to the front end. The front end sends a 'request user comment of book' to the server, which then retrieves the 'user comment of book' from the database. The server returns this comment to the front end, which then renders the 'content write page'. The user checks their comment and adds it to a text area. The user then submits the content form, which is sent to the server. The server performs an 'input check' before creating the content. Finally, the server creates the content and returns confirmation to the user.</p> |
| usecase     | 27   |
| description | <p>사용자가 그룹의 플랜에 독후감을 작성하는 경우를 나타내는 SD이다. precondition에 근거하여 사용자가 로그인 된 상태이며 해당 그룹의 멤버여야 한다. main success scenario에 근거하여 사용자가 그룹 페이지에서 플랜을 선택하고 글쓰기 버튼을 누르는 경우 시작된다. 서버는 DB로 부터 해당 그룹에 상응되는 책에 대하여 사용자가 남긴 모든 감상평을 불러와 화면에 출력한다. 사용자는 감상평을 선택하거나 글을 작성해 독후감을 완성할 수 있다. 사용자는 제출 버튼을 통해 서버에 전송할 수 있다. extension scenario에 근거하여 감상평을 작성하지 않은 상태에서 제출 버튼이 눌린 경우 예외 팝업을 띄우고 다시 작성 할 수 있게 한다.</p>   |

|             |  |
|-------------|--|
| diagram     |  <pre> sequenceDiagram     actor User     participant Web     participant Front     participant Server     participant DB      User-&gt;&gt;Web: logged in &amp;&amp; (purchase book    subscribe)     activate Web     Web-&gt;&gt;Front: request FileViewer page     activate Front     Front-&gt;&gt;User: render FileViewer page     deactivate Front     User-&gt;&gt;Front: click book read btn     activate Front     Front-&gt;&gt;User: request apply option     deactivate Front     User-&gt;&gt;Front: click read option (Appearance,page, font size,search)     activate Front     Front-&gt;&gt;User: apply option     deactivate Front     User-&gt;&gt;Front: click read option (comment write)     activate Front     Front-&gt;&gt;User: request comment write form     deactivate Front     User-&gt;&gt;Front: submit comment write form     activate Front     Front-&gt;&gt;User: render comment write form     deactivate Front     User-&gt;&gt;Front: click comment view option     activate Front     Front-&gt;&gt;User: request comment list with view option     deactivate Front     User-&gt;&gt;Server: create user comment of page     activate Server     Server-&gt;&gt;User: create user comment of page     deactivate Server     User-&gt;&gt;DB: get comment list with view option     activate DB     DB-&gt;&gt;User: comment list     deactivate DB     </pre> |
| usecase     | 20   |
| description | <p>사용자가 책을 열람하는 경우를 나타내는 SD이다. <b>preconditions</b>에 근거하여 사용자는 로그인 된 상태이며 해당 책을 대여 또는 소장한 상태이거나 이용권을 구독한 상태여야 한다. <b>extension scenario</b>에 근거하여 사용자가 로그인 되지 않은 경우 예러 팝업창을 띄워 로그인을 할 수 있게 하고 구매 또는 구독을 하지 않은 경우 버튼을 노출시키지 않는다. <b>main success scenario</b>에 근거하여 사용자가 책 상세페이지에서 책 읽기 버튼을 누르는 경우 시작된다.</p> <p>사용자가 사용자 메뉴에서 읽기 옵션(밝기 모드, 글씨 크기, 검색어 찾기)를 입력한 경우 옵션을 적용해 화면에 출력할 수 있게 한다.</p> <p>사용자가 사용자 메뉴에서 감상평 작성하기 옵션을 선택한 경우 서버는 감상평 보기 옵션에 따라 감상평 리스트를 화면에 출력할 수 있게 하고 사용자는 감상평을 작성할 수 있다.</p> <p>감상평을 작성하고 제출 버튼을 누르면 해당 책의 해당 페이지의 감상평이 서버로 전송된다.</p>   |

## 5. State machine diagram

Draw state machine diagrams for the client and the server system.  
 - Explain each state machine diagram.  
 12pt, 160%.

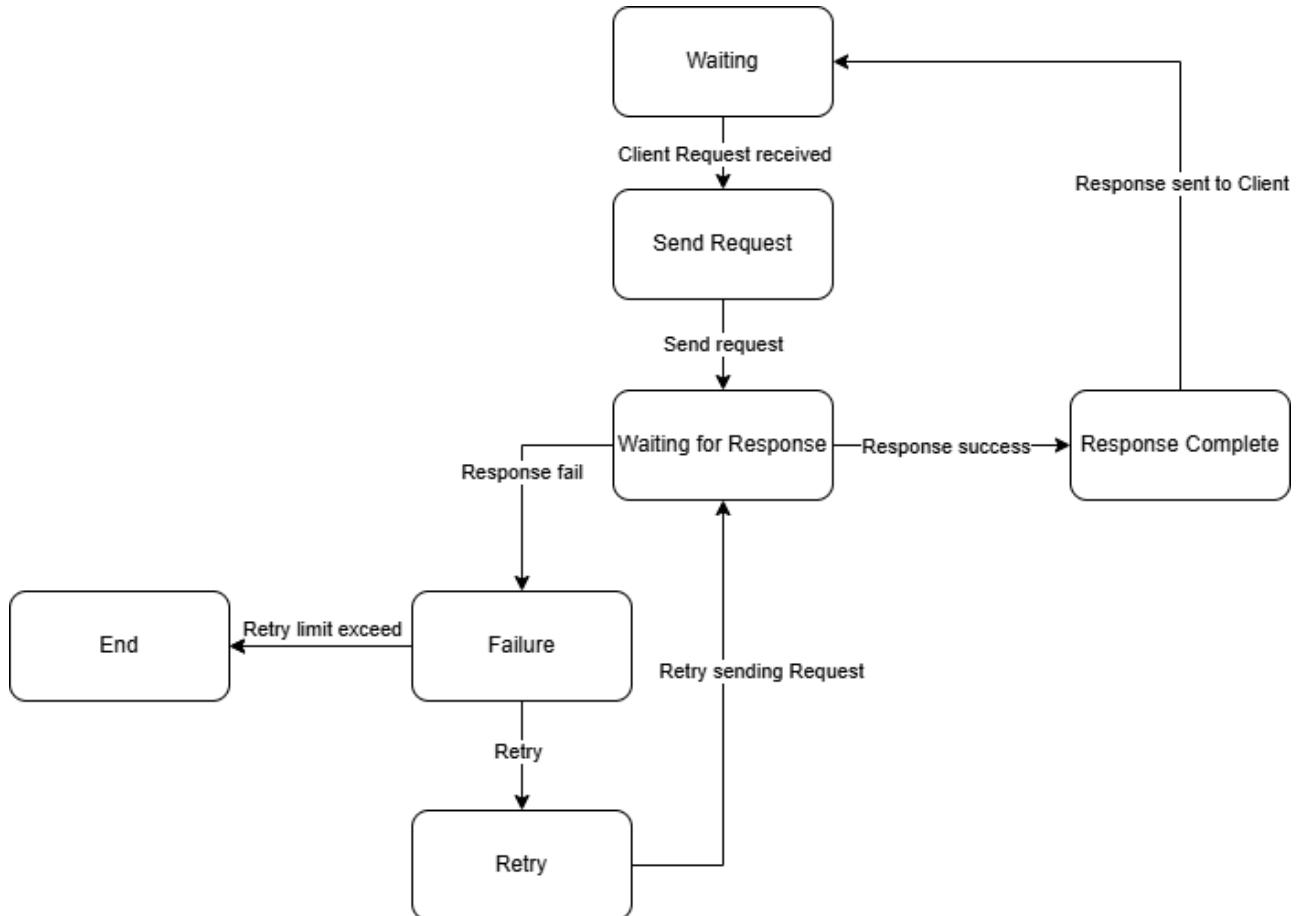
이 장은 시스템의 state machine diagram(이하 SMD)을 그리고 설명한다. [그림 5-1]은 클라이언트 측의 시스템(웹 사이트)에 관한 diagram이다.



[그림 5-1] 웹 state machine diagram

- 각 state는 사용자에게 어떤 것을 보여주고 있는지에 대한 상태를 나타내고 각 state의 이름은 CD와 1:1 대응된다.
- 본 SMD는 7개의 composite state가 있다.
- 각 화면 연결은 연결되는 composite 끼리만 전환이 가능하다.

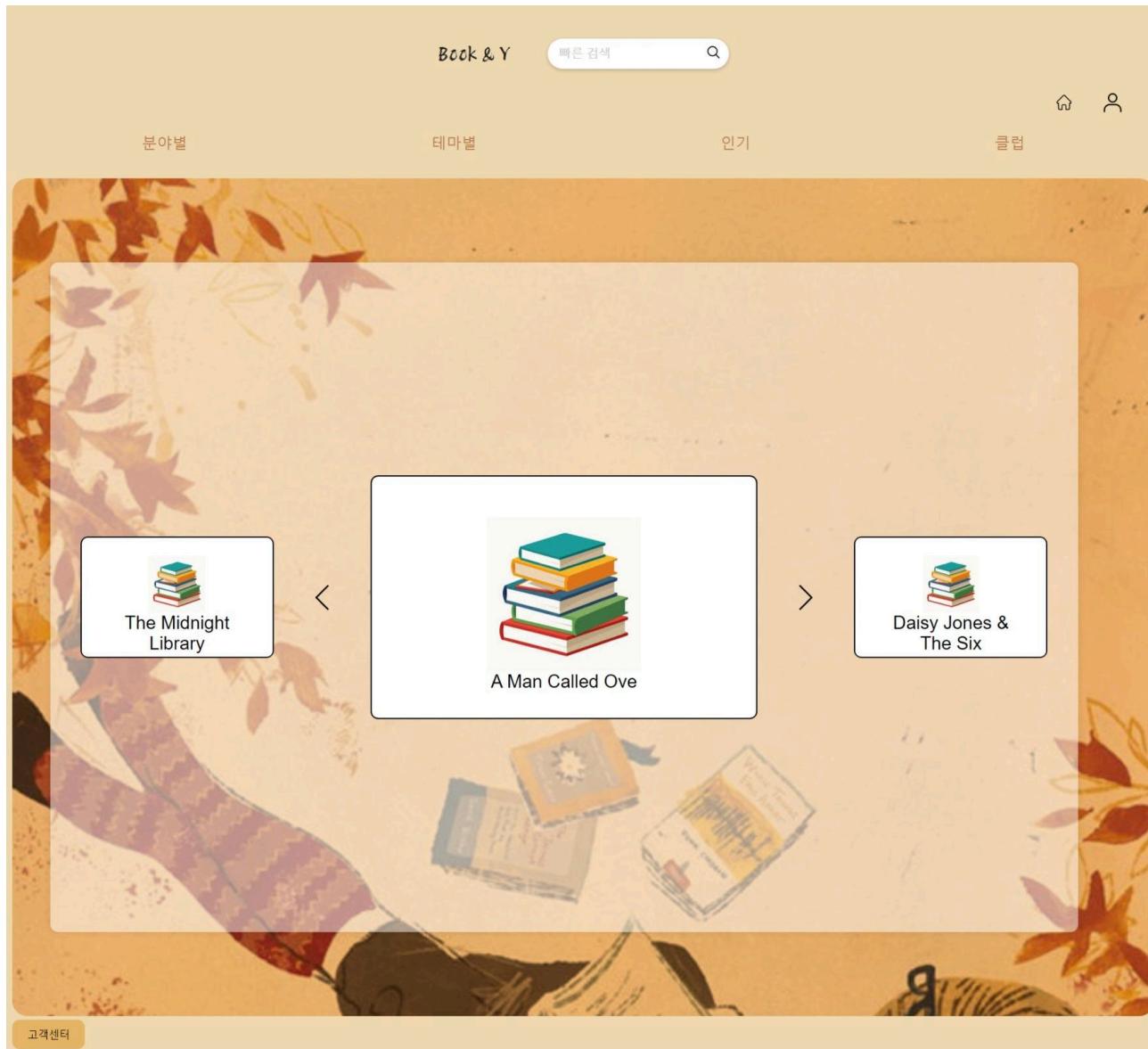
SMD 내에서는 서로 간의 수많은 요청, 응답이 발생하고 있는데 그 과정을 [그림 5-2]와 같이 확인할 수 있다.



[그림 5-2] 요청, 응답 과정 diagram

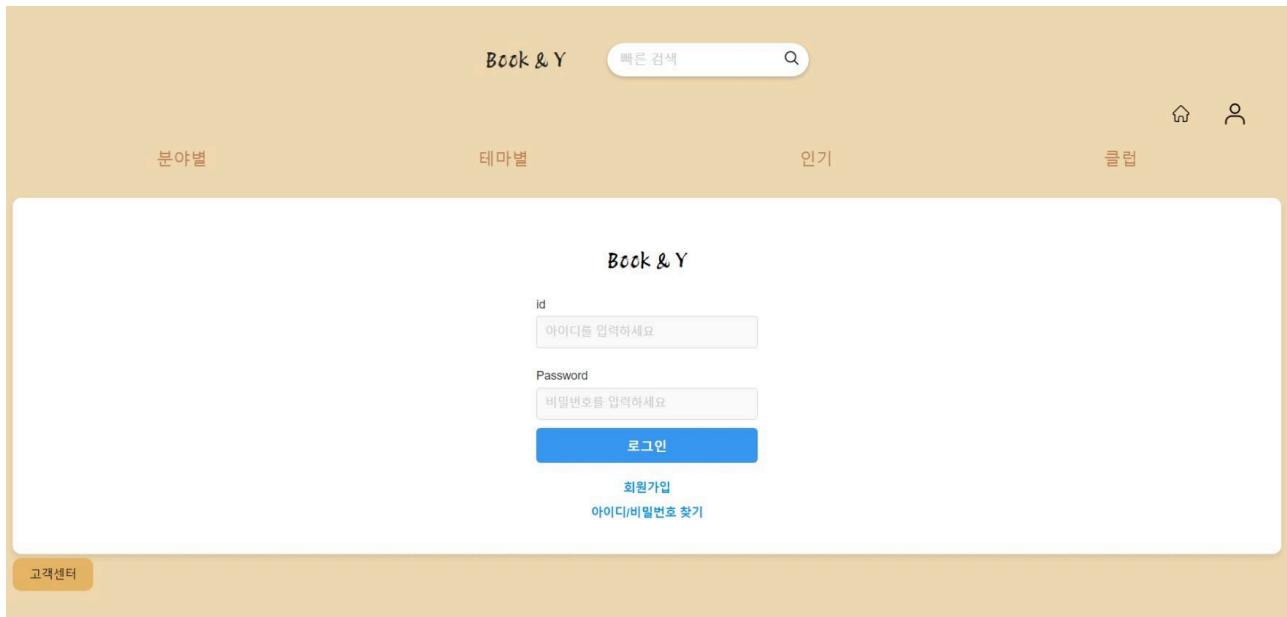
## 6. User interface prototype

- Design user interface for your software system.
  - It will be easy if you just think that you make a preliminary user manual of your system based on your user interface.
- 12pt, 160%.



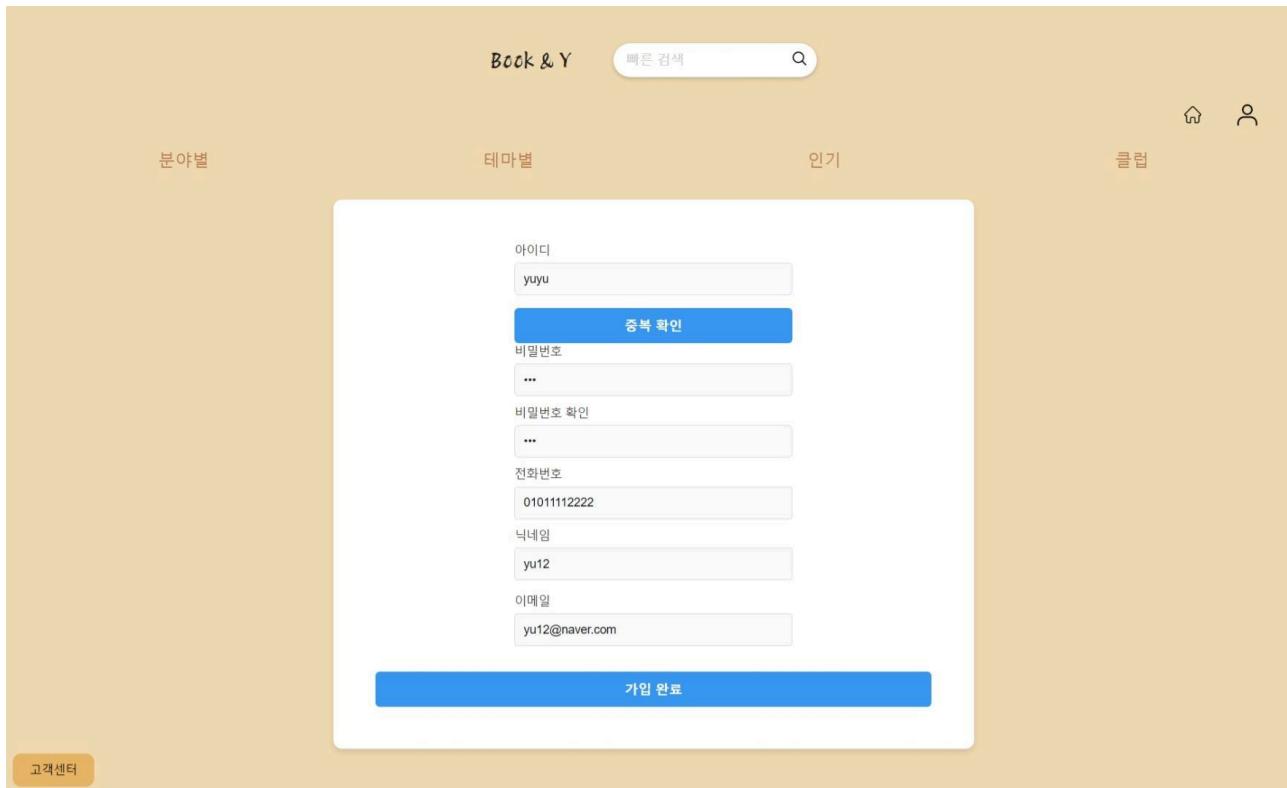
[그림 6-1] 홈 화면

위 그림은 홈 화면이다. Book & Y에 접속하면 볼 수 있으며 배너에서 분야별, 테마별, 인기순 책들과 다른 사용자들과 함께 교류할 수 있는 클럽을 확인할 수 있다. 또한 흠아이콘과 유저 아이콘을 클릭해 각각 홈화면과 마이페이지로 이동하거나 화살표를 클릭해 추천 책들을 간단히 살펴볼 수 있다.



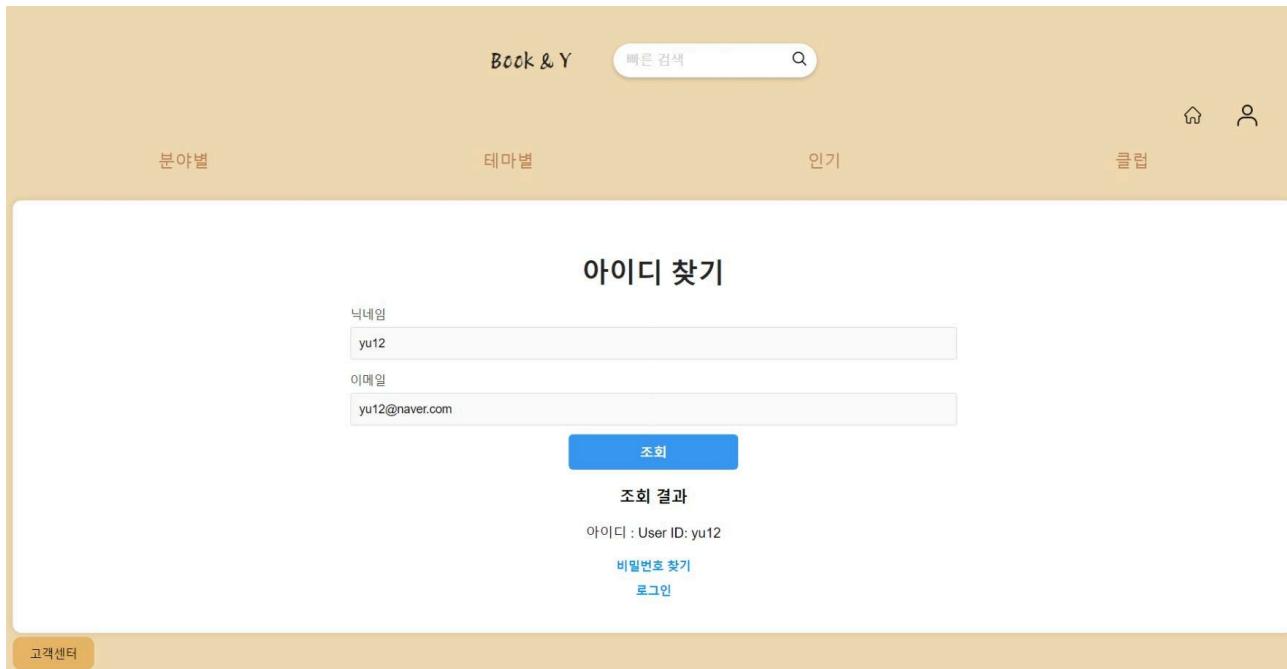
### [6-2] 로그인 화면

위 그림은 로그인 화면이다. Book & Y를 이용하기 위해 아이디, 비밀번호를 입력하여 로그인 할 수 있다. 아이디가 없다면 하단의 회원가입을 이용할 수 있으며, 아이디나 비밀번호를 잊어버렸을 때 아이디/비밀번호 찾기를 이용할 수 있다.



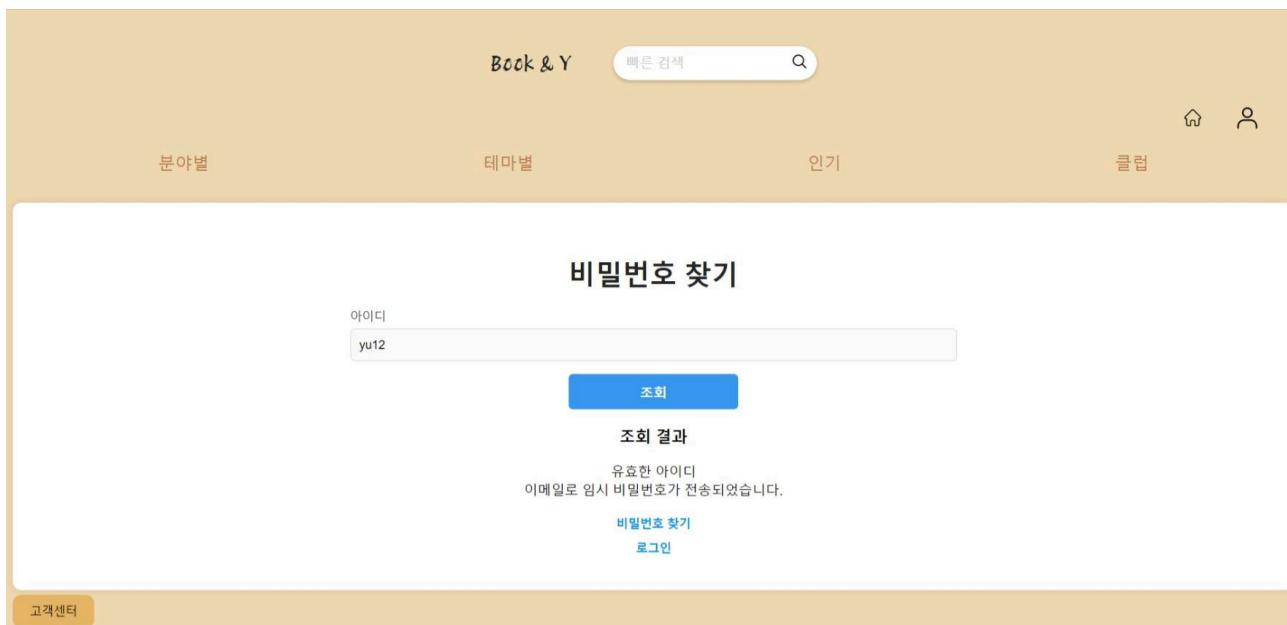
[6-3] 회원가입 화면

위 그림은 회원가입 화면이다. 아이디, 비밀번호, 비밀번호 확인, 전화번호, 닉네임, 이메일을 입력하여 회원가입을 할 수 있다. 이때 아이디는 중복 확인을 통해 중복 여부를 확인할 수 있다.



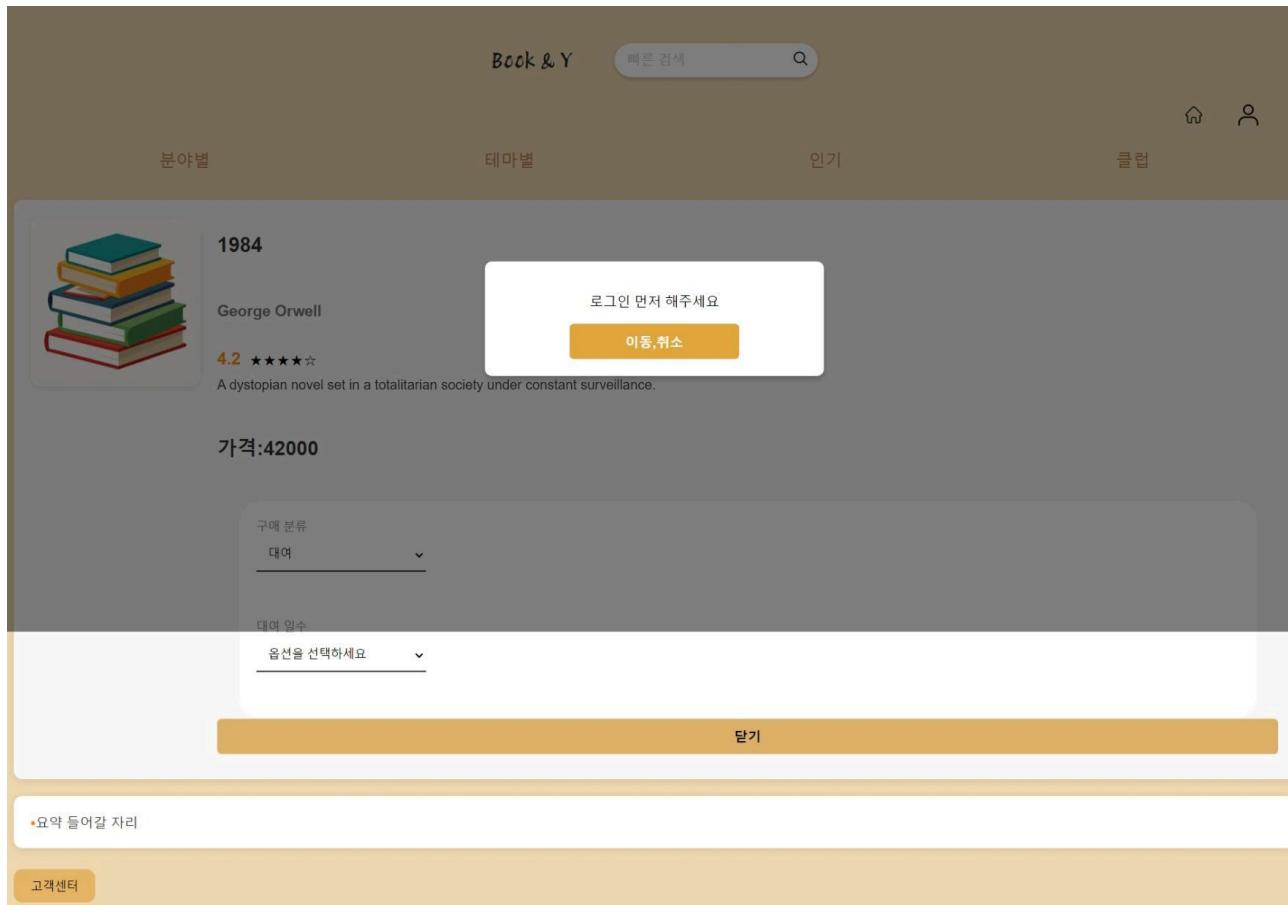
#### [6-4] 아이디 찾기 화면

위 그림은 아이디 찾기 화면이다. 본인 닉네임과 이메일을 입력 후 조회하면 그림과 같이 회원가입했던 아이디를 찾을 수 있다.



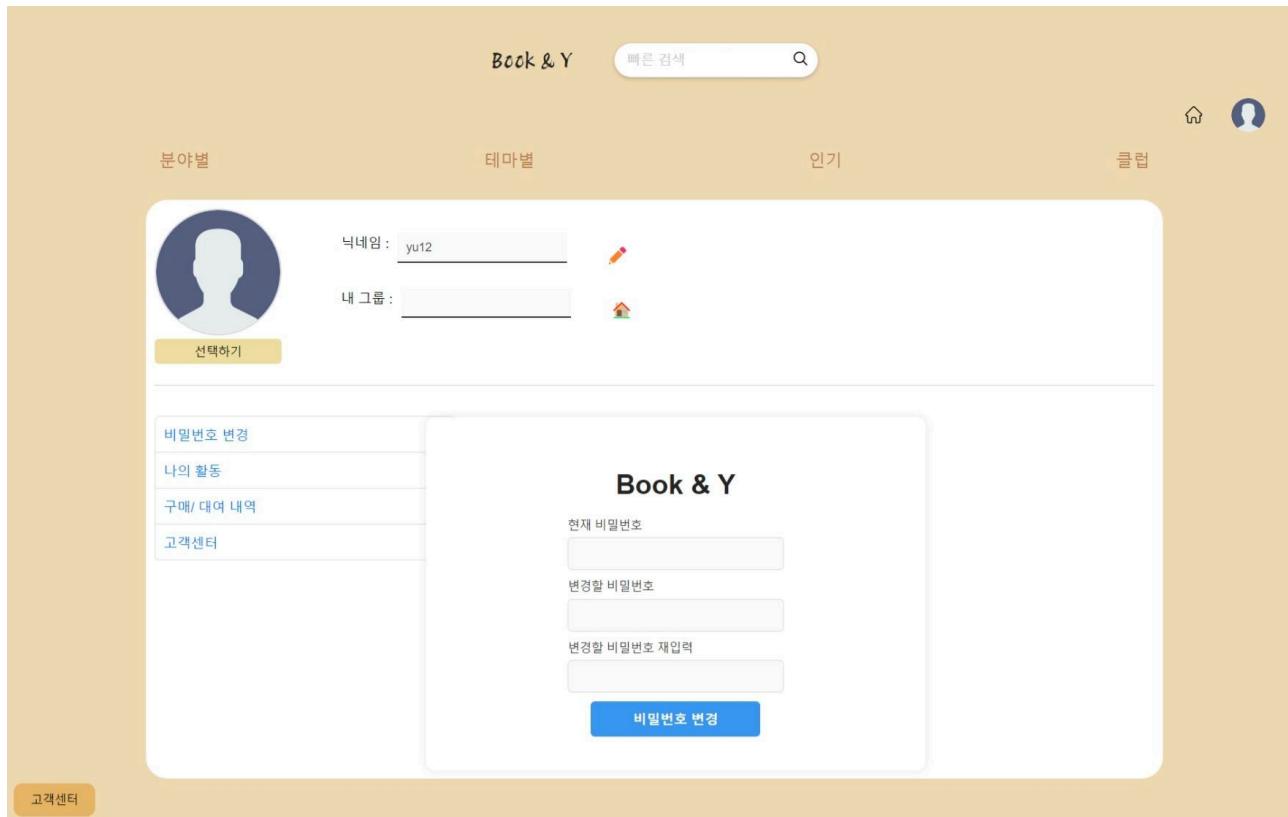
#### [6-5] 비밀번호 찾기 화면

위 그림은 비밀번호 찾기 화면이다. 본인 아이디를 입력해 조회하면 그림과 같이 조회 결과가 나온다. 또한 회원가입에 입력한 이메일로 임시 비밀번호가 전송된다.



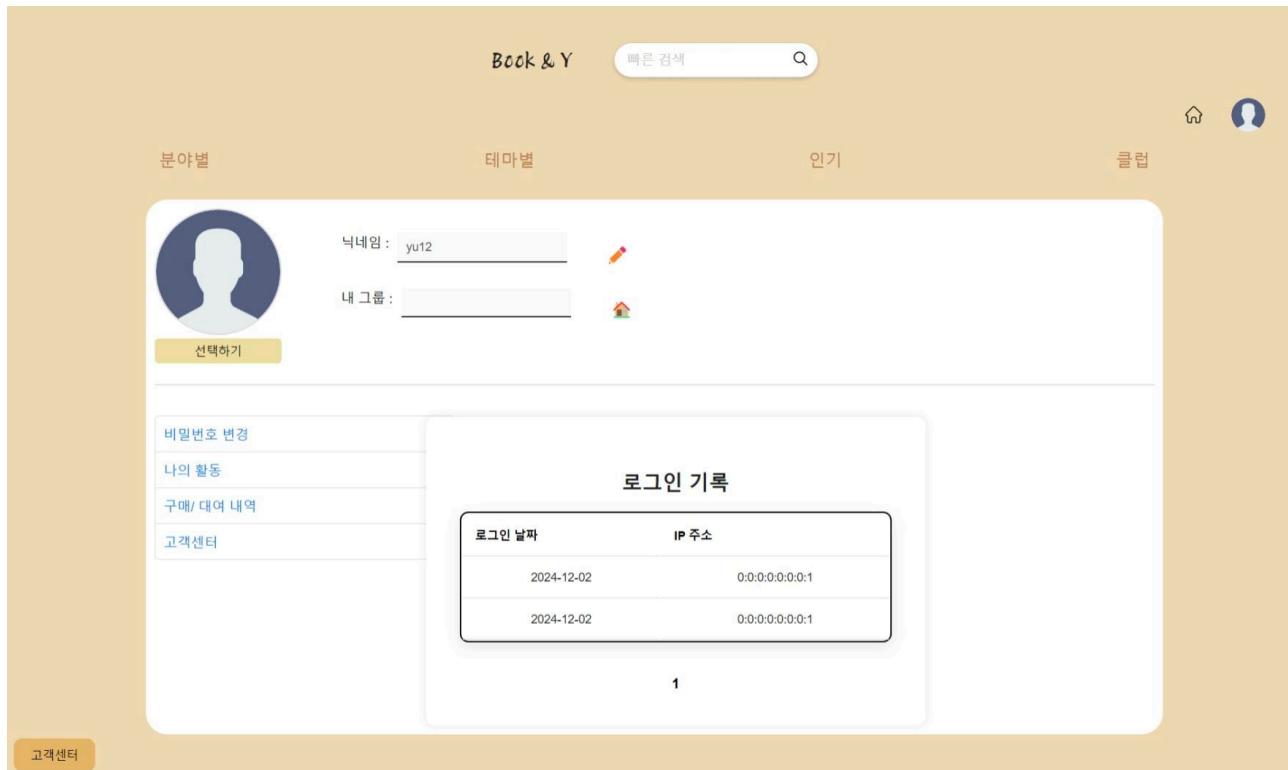
### [6-6] 접근 차단 화면

위 그림은 접근 차단 화면이다. 로그인하지 않고 책 구매나 대여를 시도할 경우, 접근을 막는 기능이다.



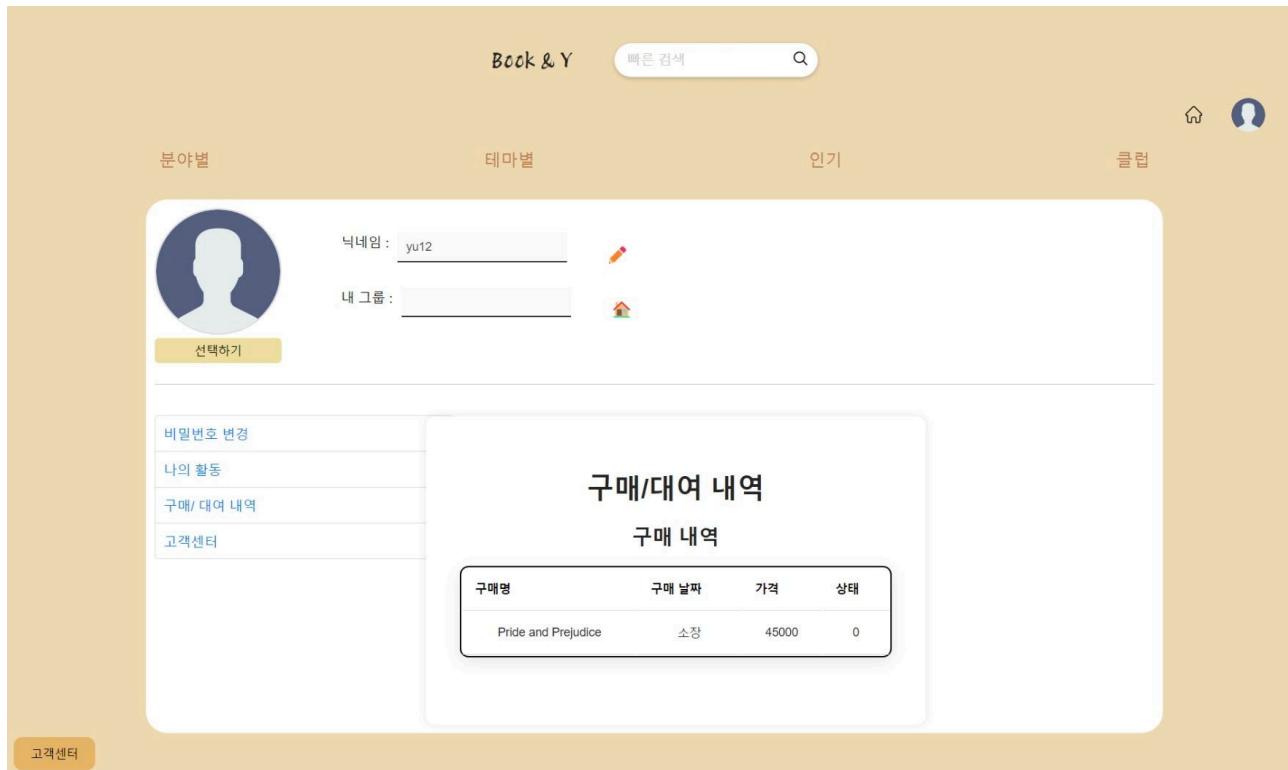
[그림 6-7] 마이페이지 및 비밀번호 변경 화면

위 그림은 마이페이지 화면이다. 프로필 사진과 내 닉네임, 내 그룹을 확인할 수 있으며 수정하기 버튼을 눌러 내가 원하는 사진으로 프로필을 수정하거나 원하는 닉네임으로 수정할 수 있다. 또한 현재 비밀번호에서 새로운 비밀번호로 변경할 수 있으며 현재 비밀번호를 입력한 후, 변경할 비밀번호를 입력·재입력 하면 비밀번호 변경이 완료된다.



[그림 6-8] 나의 활동 화면

위 그림은 나의 활동 화면이다. 과거 로그인 기록을 확인할 수 있으며 로그인 했던 날짜와 접속 IP 주소를 알 수 있다.



[그림 6-9] 구매/대여 내역 화면

위 그림은 구매/대여 내역 화면이다. 구매 내역에서는 그동안 구매한 도서명, 구매 날짜, 가격, 상태를 알 수 있다.



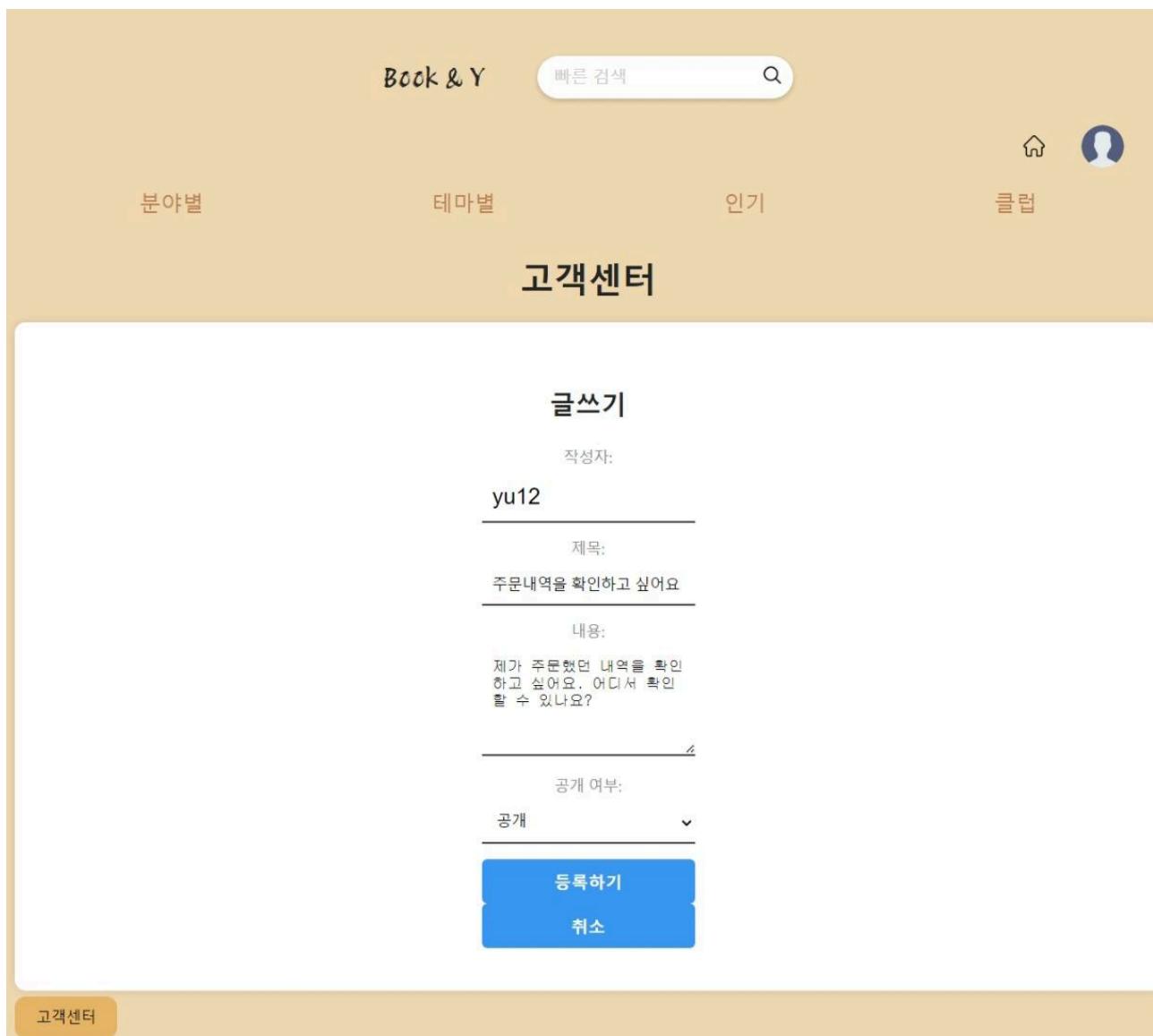
The screenshot shows the 'Guest Center' section of the university's website. At the top, there are navigation links: 'Book & Y' (highlighted in blue), '빠른 검색' (Quick Search) with a magnifying glass icon, and a user profile icon. Below these are category links: '분야별' (Category-wise), '테마별' (Theme-wise), '인기' (Popular), and '클럽' (Club). The main title '고객센터' is centered at the top of the content area. Below it is a search bar with the placeholder '검색어 입력' (Input search term) and a yellow '검색' (Search) button. A table titled 'FAQ' follows, containing one row of data:

| 번호 | 글 제목           | 작성자  | 작성일 | 상태 |
|----|----------------|------|-----|----|
| 1  | 주문내역을 확인하고 싶어요 | yu12 |     | 공개 |

Below the table is a section titled '글 등록' (Post Registration) with a yellow '글 작성하기' (Write a post) button. At the bottom left of the content area is a small orange button labeled '고객센터'.

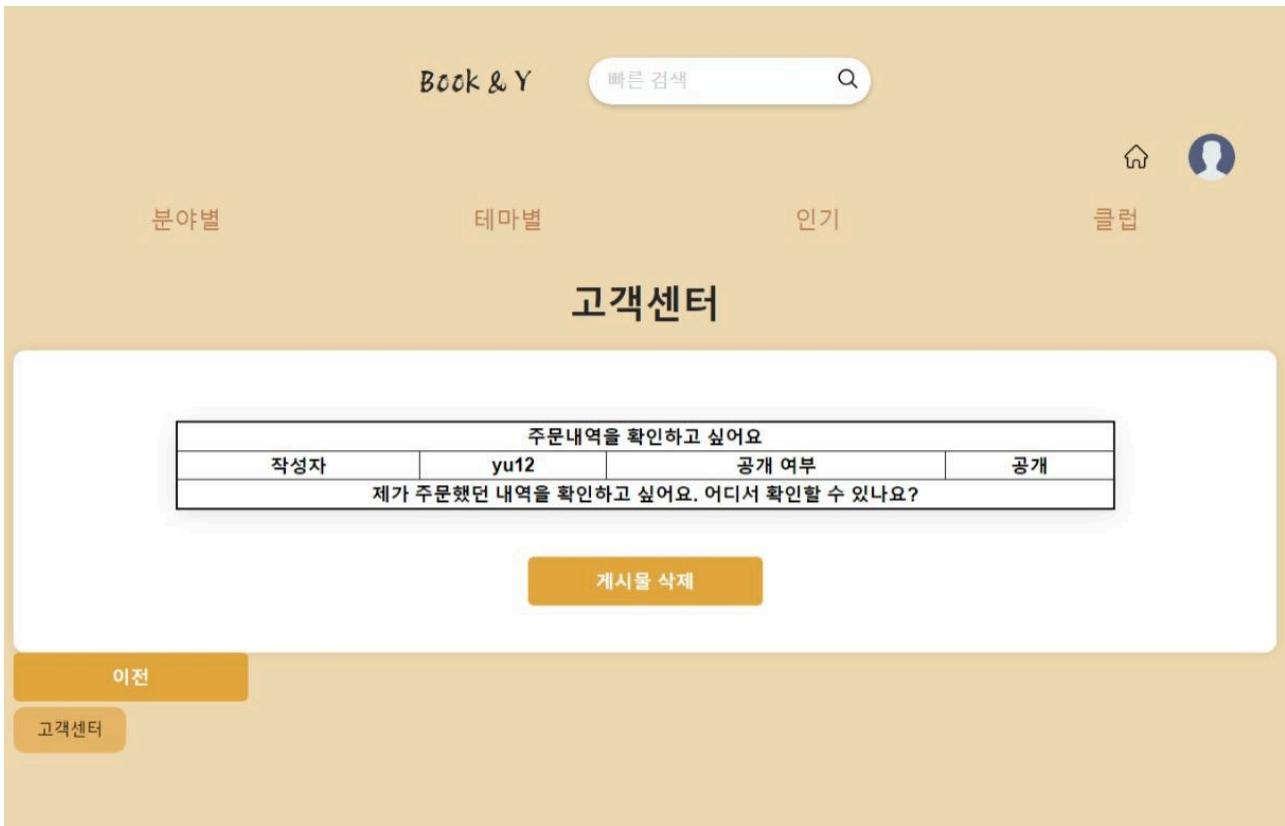
[그림 6-10] 고객센터 화면

위 그림은 고객센터 화면이다. 사용자가 직접 게시글을 쓰고 확인할 수 있으며 다른 사용자가 작성한 글을 확인할 수 있다. 또한 검색어를 입력해 게시글을 확인할 수 있다.



[그림 6-11] 글쓰기 화면

위 그림은 글쓰기 화면이다. 고객센터에서 글 등록 버튼을 눌러 글쓰기를 할 수 있다. 사용자가 제목, 내용을 입력하고 공개/비공개 여부를 선택 후, 등록하기를 누르면 글 등록이 완료된다.



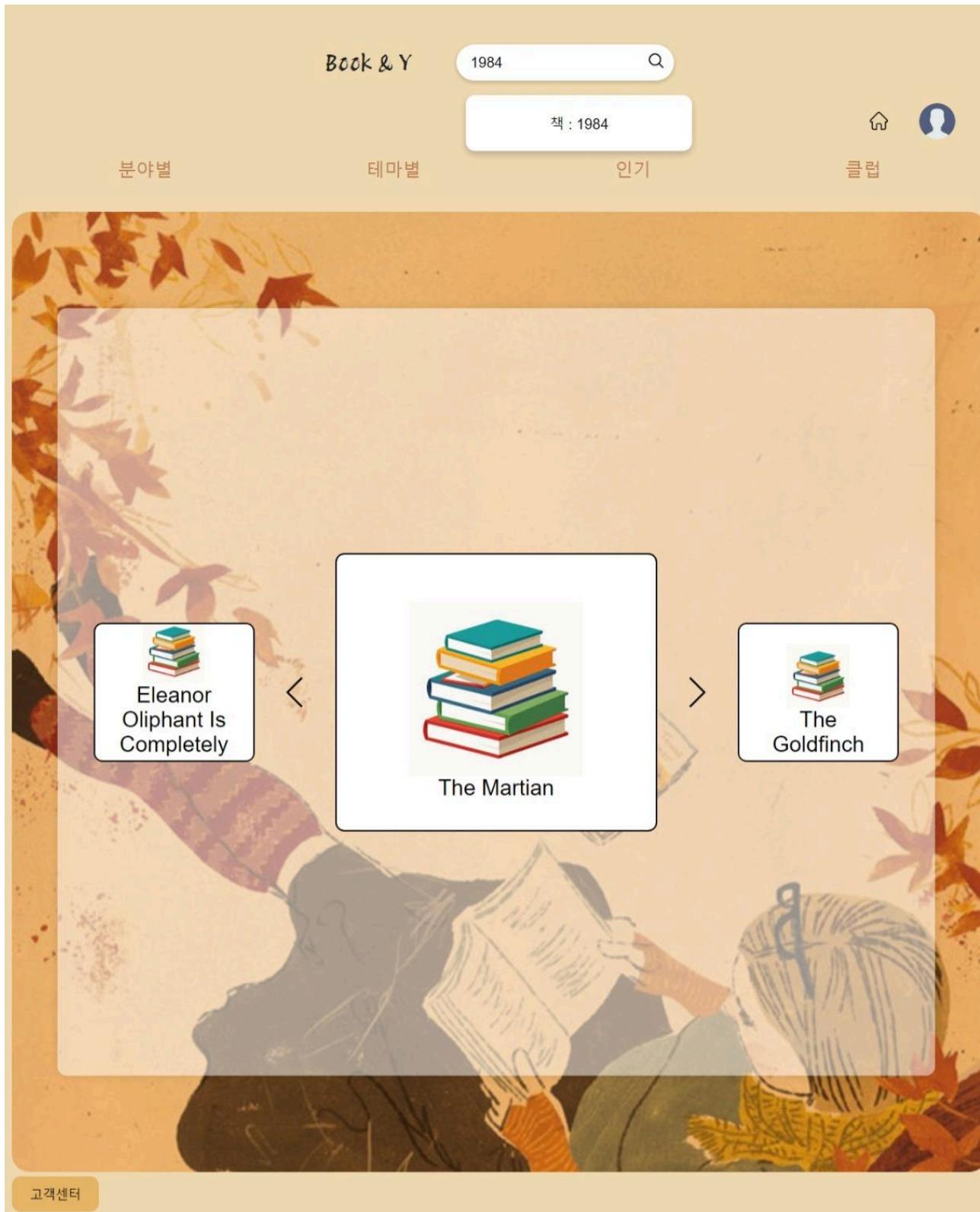
The screenshot shows a mobile application interface for 'Book & Y'. At the top, there is a search bar with the placeholder '빠른 검색' and a magnifying glass icon. To the right of the search bar is a user profile icon. Below the search bar, there are four categories: '분야별' (Category), '테마별' (Theme), '인기' (Popular), and '클럽' (Club). The main title '고객센터' (Customer Service) is centered at the top of the content area. Below it, a message card is displayed with the following content:

| 주문내역을 확인하고 싶어요                       |      |       |    |
|--------------------------------------|------|-------|----|
| 작성자                                  | yu12 | 공개 여부 | 공개 |
| 제가 주문했던 내역을 확인하고 싶어요. 어디서 확인할 수 있나요? |      |       |    |

A yellow button labeled '게시물 삭제' (Delete Post) is located below the message card. At the bottom left of the content area, there are two buttons: '이전' (Previous) and '고객센터' (Customer Service).

### [6-12] 게시글 화면

위 그림은 게시글 화면이다. 본인 혹은 다른 사용자가 등록한 게시글을 확인할 수 있다.



[6-13] 검색창 화면

위 그림은 검색창 화면이다. 검색창을 통해 흄, 마이페이지, 고객센터로 이동할 수 있으며, 책 제목을 입력해 상세 페이지로 이동할 수 있다.

Book & Y

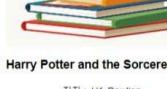
빠른 검색 검색

Home User Profile

분야별
테마별
인기
클럽

### 분야별

소설
시
역사
잡지
자격증

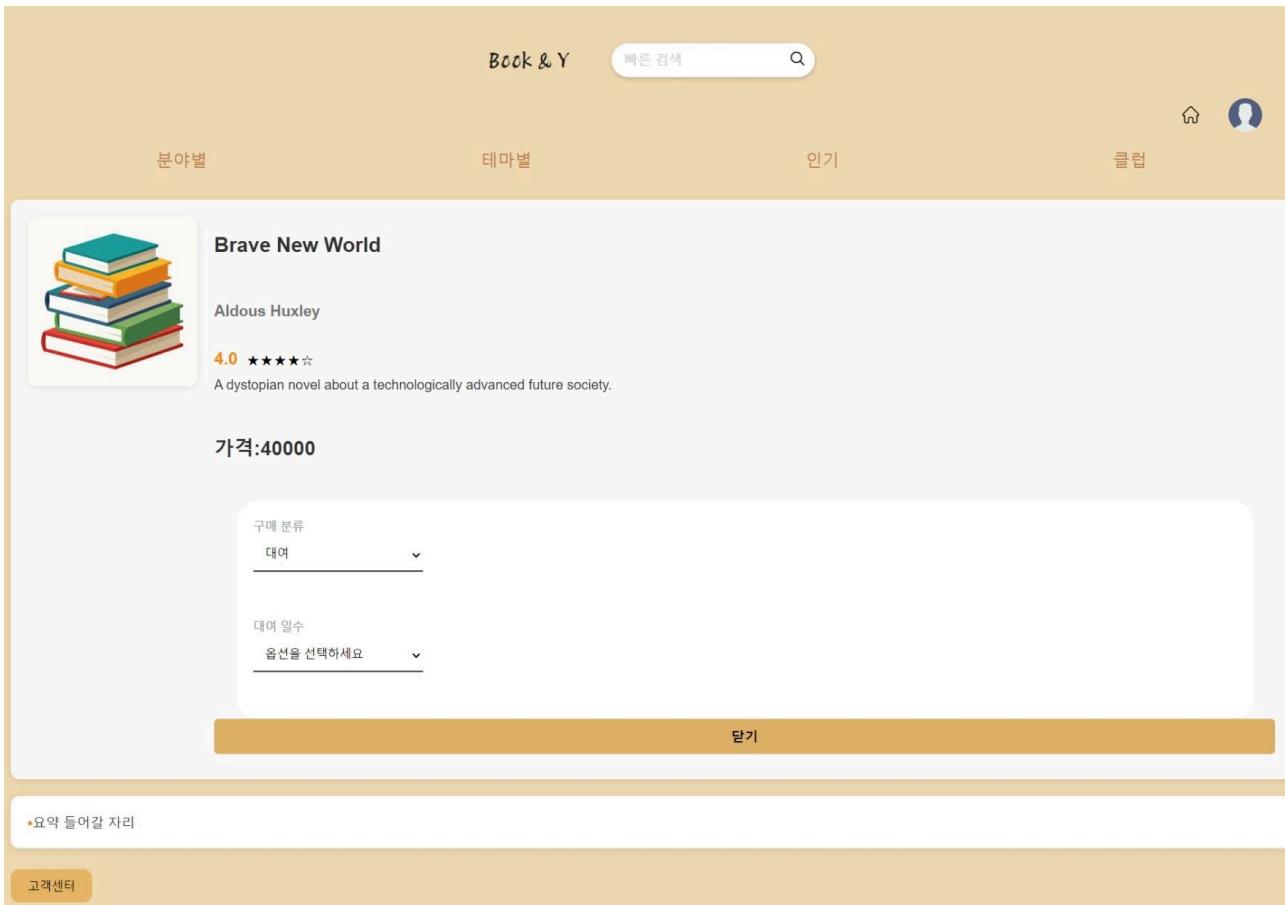
|   |   |   |   |  |
|---|---|---|---|--|
| <br><b>1984</b><br>저자 : George Orwell<br>★★★★☆                                 | <br><b>To Kill a Mockingbird</b><br>저자 : Harper Lee<br>★★★★☆                   | <br><b>The Great Gatsby</b><br>저자 : F. Scott Fitzgerald<br>★★★★½☆              | <br><b>Pride and Prejudice</b><br>저자 : Jane Austen<br>★★★★★½☆                                 | <br><b>Moby Dick</b><br>저자 : Herman Melville<br>★★★★★½☆                         |
| <b>6</b><br><br><b>The Catcher in the Rye</b><br>저자 : J.D. Salinger<br>★★★★★½☆ | <b>7</b><br><br><b>The Lord of the Rings</b><br>저자 : J.R.R. Tolkien<br>★★★★★½☆ | <b>8</b><br><br><b>Brave New World</b><br>저자 : Aldous Huxley<br>★★★★☆          | <b>9</b><br><br><b>Fahrenheit 451</b><br>저자 : Ray Bradbury<br>★★★★☆                           | <b>10</b><br><br><b>The Hobbit</b><br>저자 : J.R.R. Tolkien<br>★★★★★              |
| <b>11</b><br><br><b>Animal Farm</b><br>저자 : George Orwell<br>★★★★★☆           | <b>12</b><br><br><b>Wuthering Heights</b><br>저자 : Emily Brontë<br>★★★★☆       | <b>13</b><br><br><b>Jane Eyre</b><br>저자 : Charlotte Brontë<br>★★★★★½☆         | <b>14</b><br><br><b>The Picture of Dorian Gray</b><br>저자 : Oscar Wilde<br>★★★★★½☆            | <b>15</b><br><br><b>The Chronicles of Narnia</b><br>저자 : C.S. Lewis<br>★★★★★   |
| <b>16</b><br><br><b>The Alchemist</b><br>저자 : Paulo Coelho<br>★★★★☆          | <b>17</b><br><br><b>The Kite Runner</b><br>저자 : Khaled Hosseini<br>★★★★☆     | <b>18</b><br><br><b>The Fault in Our Stars</b><br>저자 : John Green<br>★★★★★½☆ | <b>19</b><br><br><b>Harry Potter and the Sorcerer's ...</b><br>저자 : J.K. Rowling<br>★★★★★½☆ | <b>20</b><br><br><b>The Handmaid's Tale</b><br>저자 : Margaret Atwood<br>★★★★★☆ |
| <b>21</b><br><br><b>The Bell Jar</b><br>저자 : Sylvia Plath<br>★★★★☆           | <b>22</b><br><br><b>The Road</b><br>저자 : Cormac McCarthy<br>★★★★☆            | <b>23</b><br><br><b>Gone Girl</b><br>저자 : Gillian Flynn<br>★★★★☆             | <b>24</b><br><br><b>The Great Alone</b><br>저자 : Kristin Hannah<br>★★★★★½☆                   | <b>25</b><br><br><b>Little Fires Everywhere</b><br>저자 : Celeste Ng<br>★★★★★☆  |

1 2 3 4 5

고객센터

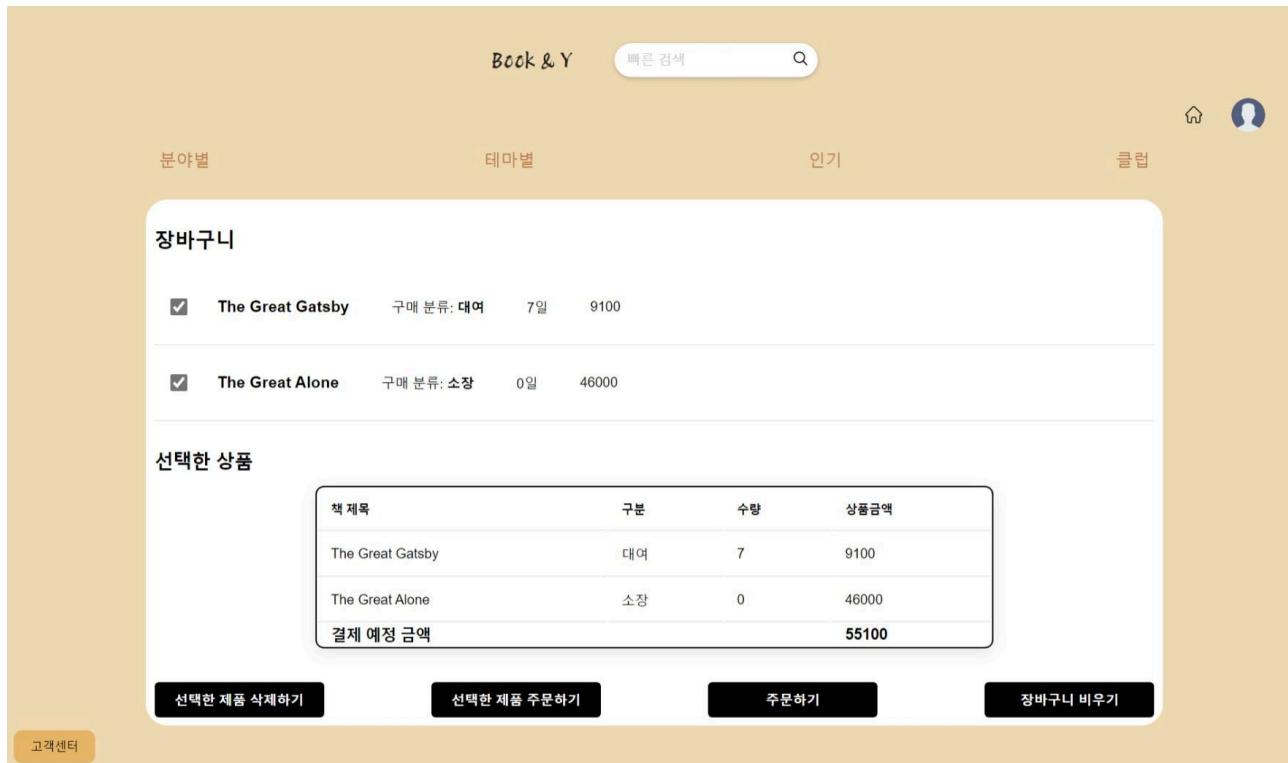
[그림 6-14] 분야별 화면

위 그림은 책 분야별 화면이다. 홈 화면에서 분야별, 테마별, 인기 배너를 클릭해 확인할 수 있으며 랭크 순으로 나열돼 내가 찾고자 하는 책을 한눈에 찾을 수 있다. 책에 커서를 올리면 간단한 줄거리를 확인할 수 있다.



[그림 6-15] 구매 화면

위 그림은 구매 화면이다. 내가 사고자 하는 책을 클릭하면 해당 책의 제목, 저자, 평점, 구매수, 리뷰, 요약을 확인할 수 있으며 구매 분류를 통해 소장 또는 대여를 선택할 수 있다. 대여를 선택할 경우 7일, 14일, 30일 중 선택할 수 있다.



The screenshot shows the Book & Y website's shopping cart page. At the top, there is a search bar with the placeholder "빠른 검색" and a magnifying glass icon. Below the search bar are four navigation links: "분야별" (Category), "테마별" (Theme), "인기" (Popular), and "클럽" (Club). In the center, there is a section titled "장바구니" (Cart) which lists two items:

|                                     |                  |           |    |       |
|-------------------------------------|------------------|-----------|----|-------|
| <input checked="" type="checkbox"/> | The Great Gatsby | 구매 분류: 대여 | 7일 | 9100  |
| <input checked="" type="checkbox"/> | The Great Alone  | 구매 분류: 소장 | 0일 | 46000 |

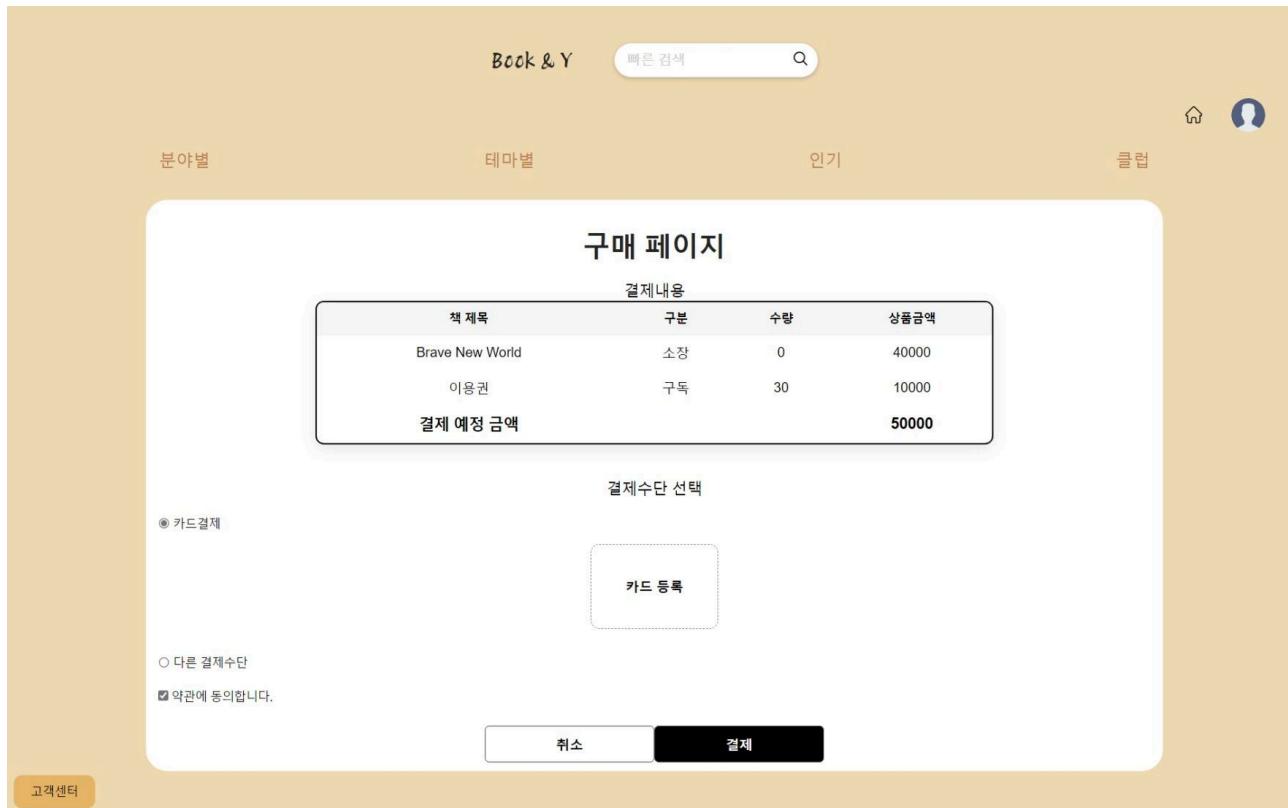
Below the cart summary, there is a section titled "선택한 상품" (Selected Products) which displays a table of the selected items:

| 책 제목             | 구분 | 수량 | 상품금액         |
|------------------|----|----|--------------|
| The Great Gatsby | 대여 | 7  | 9100         |
| The Great Alone  | 소장 | 0  | 46000        |
| 결제 예정 금액         |    |    | <b>55100</b> |

At the bottom of the cart page, there are four buttons: "선택한 제품 삭제하기" (Delete Selected Product), "선택한 제품 주문하기" (Order Selected Product), "주문하기" (Order), and "장바구니 비우기" (Empty Cart). A small "고객센터" (Customer Service) button is located at the bottom left.

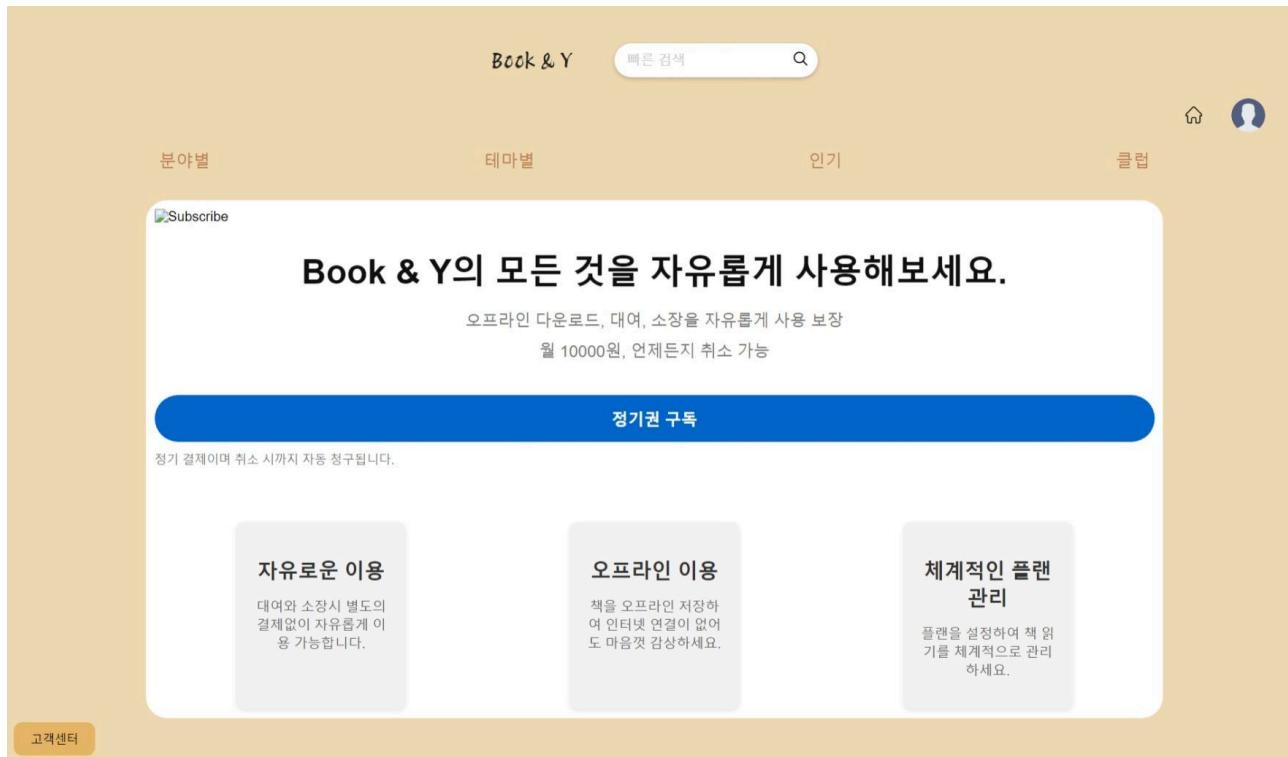
[그림 6-16] 장바구니 화면

위 그림은 장바구니 화면이다. 책 구매 화면에서 소장 혹은 대여를 선택한 뒤 장바구니 버튼을 누르면 볼 수 있으며 선택한 제품 삭제하기, 선택한 제품 주문하기, 주문하기, 장바구니 비우기 기능을 사용할 수 있다.



[그림 6-17] 결제 화면

위 그림은 결제 화면이다. 장바구니 화면에서 구매하기 버튼을 클릭하면 선택한 책을 구매할 수 있다. 결제수단으로 카드를 등록하여 결제하거나 다른 결제수단으로 결제할 수 있다.



[그림 6-18] 구독 화면

위 그림은 구독 화면이다. 대여하거나 소장하고 싶은 책이 많을 때 별도의 결제 없이 자유롭게 이용할 수 있는 구독 기능이다. 더불어 인터넷 연결 없이 오프라인으로 책을 감상할 수 있으며 그룹을 사용해 체계적으로 책 읽기를 관리할 수 있다.

Book & Y
빠른 검색
검색

분야별
테마별
인기

### 메밀꽃 필 무렵

0/호석

여름 장이란 애당초에 글려서, 해는 아직 중천에 있건만 장판은 벌써  
쓸쓸하고 더운 햇볕이 떨어 놓은 전 휘장 밑으로 등줄기를 훑훅 볶는  
다. 마을 사람들은 거지반 돌아간 뒤요, 팔리지 못한 나무꾼 패가 길  
거리에 궁싯거리고들 있으나, 석유병이나 받고 고깃마리나 사면 족  
할 이 죽들을 바라고 언제까지든지 버티고 있을 법은 없다. 출출스럽  
게 날아드는 파리 떼도 장난꾼 각다귀들도 귀찮다. 일금뱅이요 원순  
잡이인 드림전의 허 생원은 기어코 동업의 조 선달을 나 꾸어 보았  
다.“그만 거둘까?” “잘 생각했네. 봉평장에서 한 번이나 흐붓하게 사  
본 일 있었을까. 내일 대화장에 서나 한몫 벌어야겠네.” “오늘 밤은  
밤을 새서 걸어야 될걸?” “달이 뜨렷다?” 절령절령 소리를 내며 조 선  
달이 그 날 산 돈을 따지는 것을 보고 허 생원은 말뚝에서 넓은 휘장  
을 걷고 벌여 놓았던 물건을 거두기 시작하였다. 무명 필과 주단바리  
가 두 고리짝에 꽉 찼다. 명석 위에는 천 조각이 어수선하게 남았다.  
다른 죽들도 벌써 거진 전들을 걷고 있었다. 악빠르게 떠나는 패도  
있었다. 어물 장수도, 땜 장이도, 엿 장수도, 생강 장수도 끌들이 보이  
지 않았다. 내일은 진부와 대화에 장이 선다. 죽들은 그 어느 쪽으로  
든지 밤을 새워 육칠십 리 밤길을 타박거리지 않으면 안 된다. 장판  
은 잔 치 위 마당같이 어수선하게 벌어지고, 술집에서는 심음이 터져  
있었다. 주정꾼 육지거리에 섞여 계집의 양칼진 목소리가 끊어졌다.  
장날 저녁은 정해 놓고 계집의 고함소리로 시작되는 것 이다.

“생원, 시침을 떼두 다 아네…… 총죽집 말이야.” 계집 목소리로 문  
득 생각난 듯이 조 선달은 비죽이 웃는다. “화중지병(畫中之餅)이지  
연소파들을 적수로 하구야 대거리가 돼야 말이지.” “그렇지도 않을  
걸. 죽들이 사죽을 못 쓰는 것도 사실은 사실이니, 아무리 그렇다 군  
해 두 왜 그 둘이 말일세. 감쪽같이 총죽집을 후린 눈치거든.” “우어  
그 애송이가? 물건 가지고 낚었나 부지, 착실한 녀석인 줄 알았더니.”  
“그 길만은 알 수 있나…… 궁리 말구 가 보세나그려. 내 한턱 쓴세.”  
그다지 마음이 당기지 않는 것을 풋아갔다. 허 생원은 계집과는 연분  
이 멀었다. 일금뱅이 상판을 쳐들고 대어 설 솟기도 없었으나, 계집  
편에서 정을 보낸 적도 없었고, 쓸쓸하고 뒤틀린 반생이었다. 총죽  
집을 생각만 하여도 철없이 얼굴이 붉어지고 발밑이 떨리고 그 자리  
에 소스라차 버린다. 총죽집 문을 들어서 술작석에서 짜장 동이를  
만났을 때에는 어찌된 서슬엔지 발끈 화가 나 버렸다. 상 위에 붉은  
얼굴을 쳐들고 계집과 농탕치는 것을 보고서야 견딜 수 없었던  
것이다. 녀석이 제법 난질꾼인데 끌사납다. 머리에 피도 안 마른 녀  
석이 낮부터 술 처먹고 계집과 농탕이야. 장돌뱅이 망신만 시키고 돌  
아다니누나. 그 곁에 우리들과 한몫 보자 는 셈이지. 동이 앞에 악아  
서면서부터 책망이었다. 걱정도 팔자요 하는 듯이 뻔히 쳐다보는 상  
기된 눈망울에 부딪힐 때, 결김에 따귀를 허나 갑자 두지 않고는 배  
길 수가 없었다. 동이도 화를 쓰고 팩하게 일어서기는 하였으나, 허  
생원은 조금도 동색하는 법 없이 마음먹은 대로는 다 지껄였다 — 어  
디서 쫓아먹은 선머슴인지는 모르겠으나, 너에게도 아비 어미 있겠  
지. 그 사나운 꿀 보면 맘 좋겠다. 장사란 탐탁하게 해야 되지, 계집  
이 다 무어야. 나가거라, 냉큼 꿀 치워.

이전
1 / 6
다음

고객센터
사용자 메뉴

## [6-19] 전자책 읽기 화면

위 그림은 책 읽기 화면이다. 책을 소유 또는 대여했을 때, 책의 상세페이지로 이동하면 전자책 형태로 읽을 수 있다.

Book & Y
빠른 검색
검색

분야별
테마별
인기
클럽

### 메밀꽃 필 무렵

0/호석

여름 장이란 애당초에 글려서, 해는 아직 중천에 있건만 장판은 벌써  
쓸쓸하고 더운 햇볕이 떨어 놓은 전 휘장 밑으로 등줄기를 훑 훑 볶는  
다. 마을 사람들은 거지반 돌아간 뒤요, 팔리지 못한 나무꾼 패가 길  
거리에 궁싯거리고들 있으나, 석유병이나 받고 고깃마리나 사면 족  
할 이 죽들을 바라고 언제까지든지 버티고 있을 법은 없다. 출출스럽  
게 날아드는 파리 떼도 장난꾼 각다귀들도 귀찮다. 일금뱅이요 원순  
잡이인 드림전의 허 생원은 기어코 동업의 조 선달을 나 꾸어 보았  
다. “그만 거둘까?” “잘 생각했네. 봉평장에서 한 번이나 흐뭇하게 사  
본 일 있었을까. 내일 대화장에 서나 한몫 벌어야겠네.” “오늘 밤은  
밤을 새서 걸어야 될걸?” “달이 뜨렷다? 절령절령 소리를 내며 조 선  
달이 그 날 산 돈을 따지는 것을 보고 허 생원은 말뚝에서 넓은 휘장  
을 걷고 별여 놓았던 물건을 거두기 시작하였다. 무명 필과 주단바리  
가 두 고리짝에 꽉 찼다. 명석 위에는 천 조각이 어수선하게 남았다.  
다른 죽들도 벌써 거진 전들을 걷고 있었다. 악빠르게 떠나는 패도  
있었다. 어물 장수도, 땜 장이도, 엿 장수도, 생강 장수도 끌들이 보이  
지 않았다. 내일은 진부와 대화에 장이 선다. 죽들은 그 어느 쪽으로  
든지 밤을 새워 육칠십 리 밤길을 타박거리지 않으면 안 된다. 장판  
은 잔 치 위 마당같이 어수선하게 벌어지고, 술집에서는 심음이 터져  
있었다. 주정꾼 육지거리에 섞여 계집의 양칼진 목소리가 끊어졌다.  
장날 저녁은 정해 놓고 계집의 고함소리로 시작되는 것 이다.

“생원, 시침을 떼두 다 아네…… 총죽집 말이야.” 계집 목소리로 문  
득 생각난 듯이 조 선달은 비죽이 웃는다. “화중지병(畫中之餅)이지  
연소파들을 적수로 하구야 대거리가 돼야 말이지.” “그렇지도 않을  
걸. 죽들이 사죽을 못 쓰는 것도 사실은 사실이니, 아무리 그렇다 군  
해 두 왜 그 둘이 말일세. 감쪽같이 총죽집을 후린 눈치거든.” “우어  
그 애송이가? 물건 가지고 낚었나 부지, 착실한 녀석인 줄 알았더니.”  
“그 길만은 알 수 있나…… 궁리 말구 가 보세나그려. 내 한턱 쓸세.”  
그다지 마음이 당기지 않는 것을 풋아갔다. 허 생원은 계집과는 연분  
이 멀었다. 일금뱅이 상판을 쳐들고 대어 설 솟기도 없었으나, 계집  
편에서 정을 보낸 적도 없었고, 쓸쓸하고 뒤틀린 반생이었다. 총죽  
집을 생각만 하여도 철없이 얼굴이 붉어지고 발밑이 떨리고 그 자리  
에 소스라차 버린다. 총죽집 문을 들어서 술작석에서 짜장 동이를  
만났을 때에는 어찌된 서슬엔지 발끈 화가 나 버렸다. 상 위에 붉은  
얼굴을 쳐들고 제법 계집과 농탕치는 것을 보고서야 견딜 수 없었던  
것이다. 녀석이 제법 난질꾼인데 끌사납다. 머리에 피도 안 마른 녀  
석이 낮부터 술 처먹고 계집과 농탕이야. 장돌뱅이.”

이전
1 / 6
다음

페이지 이동
키워드 검색
감상평 작성
다크모드
화면 확대
화면 축소

닫기

[고객센터](#)

## [6-20] 사용자 메뉴 화면

위 그림은 사용자 메뉴 화면이다. 책을 읽으면서 사용자 메뉴를 통해 페이지 이동, 특정 키워드 검색, 감상평 작성, 다크모드, 화면 확대, 화면 축소 기능을 이용할 수 있다.

Book & Y
빠른 검색
검색

분야별
테마별
인기
클럽

조 선달과 동이는 각각 제 나귀에 안장을 얹고 짐을 싣기 시작하였다. 헤가 꽤 많이 기울어 진 모양이었다. 드팀전 장들이를 시작한 지 이십 년이나 되어도 허 생원은 봉평장을 빼놓은 적은 드물었다. 총주 제전 등의 이웃 군에도 가고 멀리 영남 지방도 헤매기는 하였으나 강릉쯤에 물건 하러 가는 외에는 처음부터 끝까지 군내를 돌아다녔다.

갓새만금석의 장날에는 달보라도 확실하게 면에서 면으로 건너간다. 고향이 청주라고 자랑삼아 말하였고 고향에 돌보려 간 일도 있는 것 같지는 않았다. 장에서 장으로 가는 길의 아름다운 강산이 그대로 그에게는 그리운 고향이었다. 반날 동안이나 두번두번 걷고 장터 또는 마을에 거지반 가까웠을 때, 거친 나귀가 한바탕 우렁차게 울면 — 더구나 그것이 저녁녘이어서 등불들이 어둠 속에 깜빡거릴 무렵 이면 늘 당하는 것이건만 허 생원은 변치 않고 언제든지 가슴이 뛰놀았다. 절은 시절에는 알뜰하게 벌여 들품이나 모아 본 적도 있기는 있었으나, 읍내에 백중이 열린 해 호탕스럽게 놓고 특전을 하고 하여 사흘 동안에 다 털어 버렸다. 나귀까지 팔게 된 판이었으나 애끊는 정분에 그것만은 이를 물고 단념하였다. 결국 도로 아미타불로 장돌 이를 다시 시 작할 수밖에는 없었다. 짐승을 대리고 읍내를 도망해 나왔을 때에는 너를 팔지 않기 다행이었다. 길가에서 올면서 짐승의 등을 어루만졌던 것이었다. 빛을 지기 시작하니 재산을 모을 염은 당초에 들리고, 간신히 입에 풀칠을 하러 장에서 장으로 돌아다니게 되었다. 호탕스럽게 놓았다고는 하여도 계집 하나 후려 보지는 못하였다. 계집이란 좀 쌀쌀하고 매 정한 것이었다. 평생 인연이 뛰는 것이라고 신세가 서글퍼졌다. 일신에 가까운 것이라고는 언 제나 변함없는 한 필의 당나귀였다. 그렇다고는 하여도 꼭 한 번의 첫 일을 잊을 수는 없었다. 뒤에도 처음에도 없는 단 한 번의 괴이한 인연! 봉평에 다니기 시작한 절은 시절의 일이었으나 그것을 생각할 적만은 그도

산 보람을 느꼈다. “달밤이었으나 어떻게 해서 생각해두 도무지 알 수 없어.” 허 생원은 오늘 밤도 집어내려는 것이다. 조 선달은 친구가 된 이래 들어 왔다. 그렇다고 싶증을 낼 수도 없었으나 하고 되풀이 할 대로는 되풀이하고야 말았다. “달밤 격에 맞거든.” 조 선달 편을 버리는 보았으나, 물라 달빛에 감동하여서였다. 이지려는 있으나 보통 부드러운 빛을 흐逻이 흘리고 있다. 대화까지는 개를 둘이나 넘고 개울을 하나 건너고 별판과 산길은 지금 긴 산허 리에 걸려 있다. 밤중을 지난 고요한 속에서 짐승 같은 달의 숨소리가 손에 잡포기와 옥수수 잎새가 한층 달에 푸르게 젖었다.

밭이어서 퍼기 시작한 꽃이 소금을 뿐린 듯이 흐붓한 달빛에 숨이 막힐 지경이다. 붉은 대궁이 향기같이 애잔하고 나귀들의 걸음도 시원하다. 길이 좁은 깍닭에 세 사람은 나귀를 타고 외출로 늘어섰다. 방울 소리가 시원스럽게 딸랑딸랑 메밀밭께로 흘러간다. 앞장선 허 생원의 이야기 소리는 풍무니에 선 동이에게는 화적히는 안 들렸으나, 그는 그 대로 개운한 제 멋에 적적하지는 않았다. “장 선 꼭 이런 날밤이었네. 객줏집 토방이란 무더워서 잠이 들어야지. 밤중은 돼서 혼자 일어나 개울가에 목욕하러 나갔지. 봉평은 지금이나 그제나 마찬가지 나 보이는 곳마 다 메밀밭이어서 개울가가 어디 없이 하얀 꽃이야. 돌밭에 벚어도 좋을 것을, 달이 너무도 밝은 깨닭에 옷을 벗으려 물방앗간으로 들어가지 않았나. 이상한 일도 많지. 거기서 난데

이전
3 / 6
다음

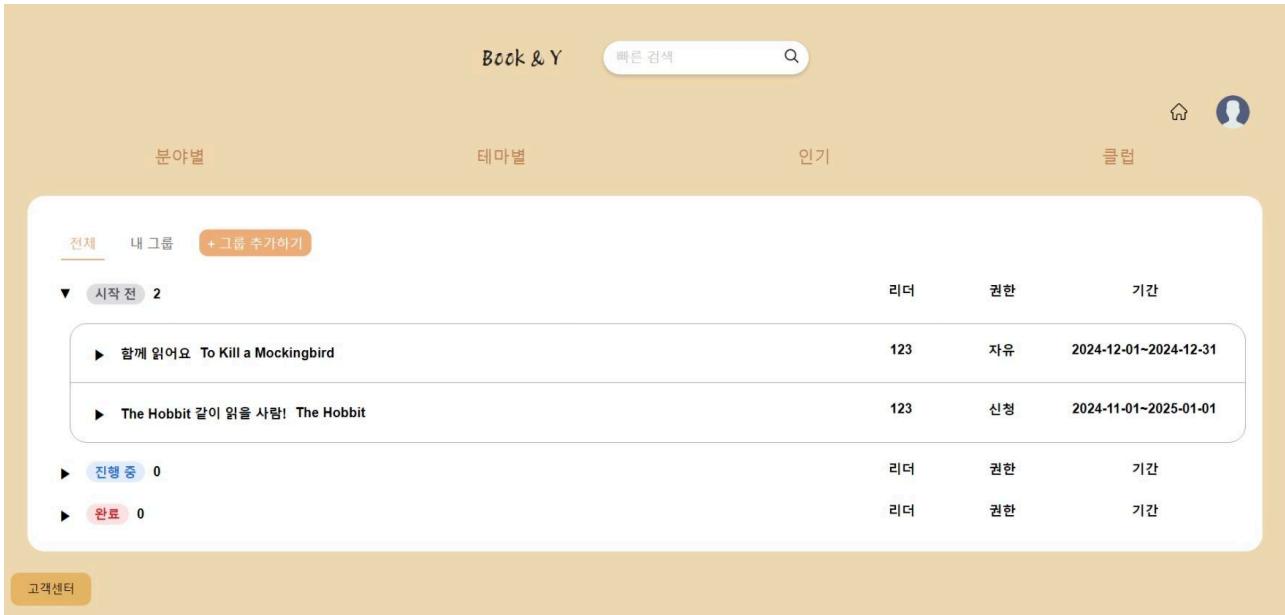
감상평 작성
감상평 입력

저장
←

yuu  
주인공의 성격이 간결하면서도 강렬하였다.
닫기

### [6-21] 감상평 작성 화면

위 그림은 감상평 작성 화면이다. 책을 읽으면서 감명깊었거나 기억에 남는 부분이 있다면 감상평을 작성할 수 있다. 또한 작성된 감상평은 참가한 그룹에도 보여져 다른 그룹 참가자들과 소통하거나 교류할 수 있다.



분야별 테마별 인기 클럽

전체 내 그룹 + 그룹 추가하기

▼ 시작 전 2

- ▶ 함께 읽어요 To Kill a Mockingbird
- ▶ The Hobbit 같이 읽을 사람! The Hobbit

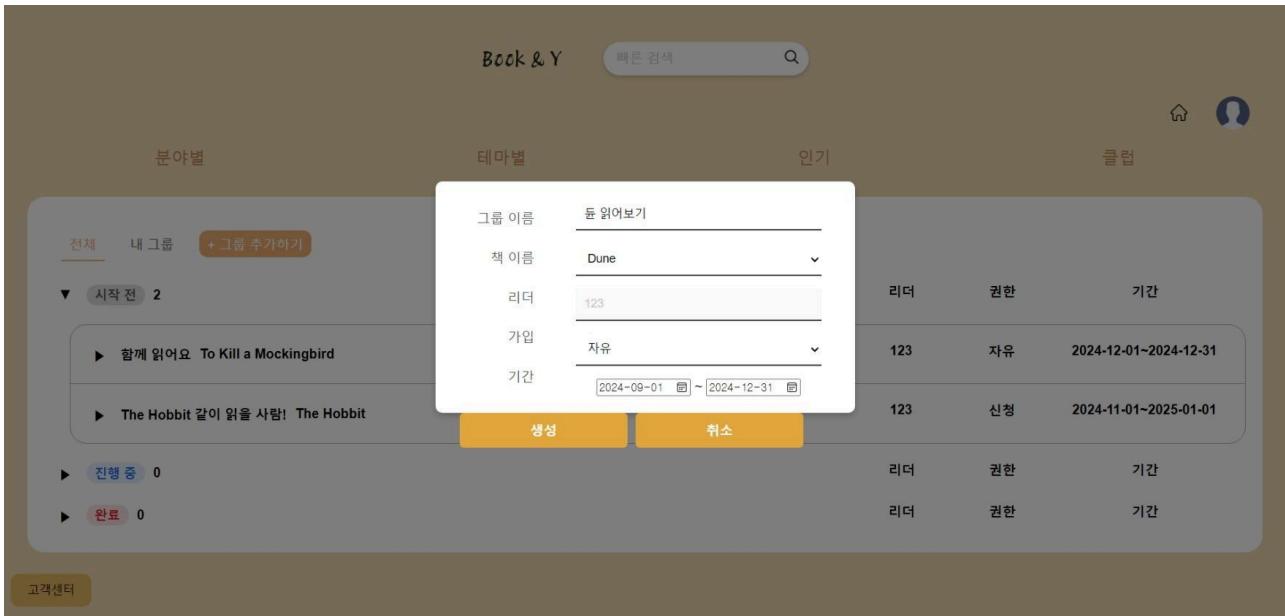
▶ 진행 중 0

▶ 완료 0

고객센터

### [6-22] 그룹 화면

위 그림은 그룹 화면이다. 그룹 페이지에서는 그림과 같이 시작 전, 진행 중, 완료된 그룹들을 확인할 수 있다.



분야별 테마별 인기 클럽

전체 내 그룹 + 그룹 추가하기

▼ 시작 전 2

- ▶ 함께 읽어요 To Kill a Mockingbird
- ▶ The Hobbit 같이 읽을 사람! The Hobbit

▶ 진행 중 0

▶ 완료 0

그룹 이름: 듣 읽어보기  
책 이름: Dune  
리더: 123  
가입: 자유  
기간: 2024-09-01 ~ 2024-12-31

생성 취소

고객센터

### [6-23] 그룹 생성 화면

위 그림은 그룹 생성 화면이다. 사용자는 그룹 추가하기를 클릭해 그룹 이름, 책 이름, 가입 방식, 기간을 입력하여 그룹을 생성할 수 있다.



The screenshot shows the Book & Y application interface. At the top, there is a search bar with the placeholder "빠른 검색" and a magnifying glass icon. Below the search bar are four navigation tabs: "Book & Y" (selected), "분야별" (Subject), "테마별" (Theme), "인기" (Popular), and "클럽" (Club). On the right side of the header are icons for a user profile and a bell.

In the main content area, there is a section titled "함께 읽어요" (Read together) with a "관리하기" (Manage) button. Below this is a table showing details for a reading session:

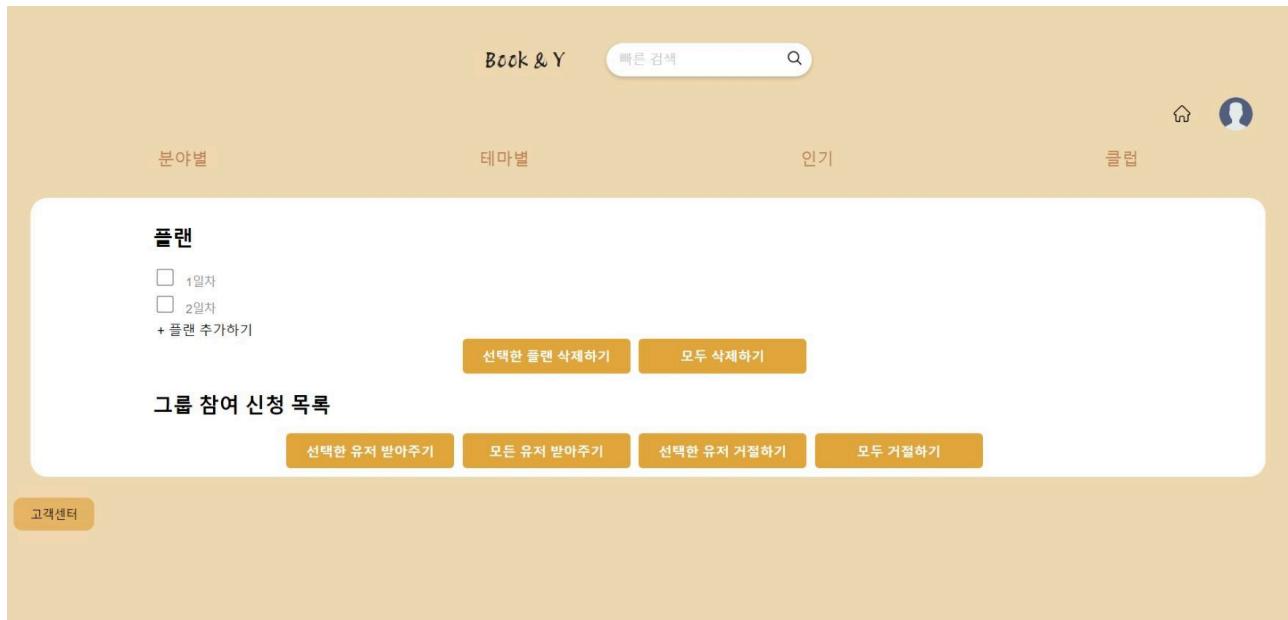
|     |                       |
|-----|-----------------------|
| 책   | To Kill a Mockingbird |
| 리더  | 123                   |
| 참여자 | 1                     |
| 가입  | 자유                    |
| 상태  | 시작 전                  |
| 기간  | 2024-12-01~2024-12-31 |

Below the table are buttons for "1일차" (Day 1) and "2일차" (Day 2), and a link "+ 플랜 추가하기" (Add plan). A section titled "그룹원 책 열람 내역" (Group member book reading history) is shown with a table header row containing "번호" (Number), "사용자" (User), and "페이지" (Page).

At the bottom left of the main content area is a "고객센터" (Customer Service) button. The background of the entire screen is a light beige color.

### [6-24] 그룹 상세 화면

위 그림은 그룹 상세 화면이다. 본인이 생성하거나 참가한 그룹의 상세 정보를 알 수 있다. 그룹 참가자는 플랜을 만들어 독서를 계획적으로 할 수 있으며 그룹원들의 책 열람 내역 또한 볼 수 있다.



The screenshot shows the Book & Y mobile application interface. At the top, there is a search bar with the placeholder "빠른 검색" and a magnifying glass icon. Below the search bar are four navigation tabs: "분야별" (Category), "테마별" (Theme), "인기" (Popular), and "클럽" (Club). On the right side of the header is a user profile icon.

In the main content area, there is a section titled "플랜" (Plan) which includes three checkboxes: "1일차", "2일차", and "+ 플랜 추가하기". Below these checkboxes are two buttons: "선택한 플랜 삭제하기" (Delete selected plan) and "모두 삭제하기" (Delete all).

Underneath the "플랜" section is a "그룹 참여 신청 목록" (Group participation application list) section. It contains four buttons: "선택한 유저 받아주기" (Receive selected users), "모든 유저 받아주기" (Receive all users), "선택한 유저 거절하기" (Reject selected users), and "모두 거절하기" (Reject all users).

A small "고객센터" (Customer Service) button is located at the bottom left of the main content area.

### [6-25] 그룹 관리 화면

위 그림은 그룹 관리 화면이다. 그룹을 생성한 참가자만이 관리할 수 있으며 플랜 삭제, 그룹 참여 신청 목록 관리를 할 수 있다.

## 7. Implementation requirements

– Describe operating environments to implement your system.  
12pt, 160%.

### 기술 스택

- 프론트엔드: React
- 백엔드: Spring Boot (Java)
- 데이터베이스: MySQL
- 서버: Node.js (API Gateway 혹은 Static File Serving)
- 기타: RESTful API 설계, JWT 인증, Axios (HTTP 클라이언트)

### 기능 요구사항

- 사용자 인증 및 권한 관리 (로그인, 회원가입)
- 데이터 CRUD(Create, Read, Update, Delete) 기능
- 실시간 데이터 업데이트 (WebSocket 또는 Polling)
- 반응형 웹 디자인 (모바일 및 데스크탑 호환)
- 에러 처리 및 로깅 기능

### 개발 환경

- IDE: IntelliJ IDEA 또는 VSCode
- 버전 관리: Git
- 빌드 도구: Maven 또는 Gradle (Spring), npm 또는 Yarn (React)

### 서버 사양(최소)

- CPU: 2코어 이상
- RAM: 4GB 이상
- 디스크: 20GB 이상의 SSD
- 운영체제: Ubuntu 20.04 이상 또는 Windows Server 2019 이상

### 클라이언트 사양(최소)

- CPU: 듀얼코어 이상
- RAM: 4GB 이상
- 디스크: 10GB 이상의 여유 공간
- 브라우저: 최신 버전의 Chrome, Firefox, Safari 등

### 네트워크

- 최소 1Mbps 이상의 인터넷 속도

## 8. Glossary

– Specifically describe all of the terms used in this documents.  
12pt, 160%.

| 이름             | 설명  |
|----------------|---|
| HTTP 상태 코드     | 특정 HTTP 요청이 성공적으로 완료되었는지 알려주는 코드. 정보를 제공하는 응답, 성공적인 응답, 리다이렉트, 클라이언트 에러, 서버 에러의 다섯 가지 그룹으로 나누어진다. |
| ResponseEntity | Spring MVC에서 HTTP 요청에 대한 응답을 제어하는 데 사용하는 클래스.   |
| List           | 자바에서 제공하는 데이터 구조 중 하나로, 여러 개의 데이터를 저장할 수 있는 자료형이다. 배열과 다르게 크기가 가변적이다.                             |
| HashMap        | Java의 컬렉션 프레임워크 중 하나로, 키(Key)와 값(Value)의 쌍으로 데이터를 저장한다. 키는 중복을 허용하지 않는다.                          |
| Optional       | null이 올 수 있는 값을 감싸는 Wrapper 클래스.  |
| DTO            | 데이터 전송 객체(Data Transfer Object)로, 계층간 데이터 교환을 위해 사용하는 객체이다.                                       |
| IP             | 인터넷에 연결되어 있는 모든 장치들(컴퓨터, 서버 장비, 스마트폰 등)을 식별할 수 있도록 각각의 장비에게 부여되는 고유 주소이다.                         |
| navigate       | React Router에서 제공하는 함수로, 특정 경로로 프로그래밍 방식의 라우팅(페이지 이동)을 수행합니다.                                     |
| useEffect      | React의 Hook으로, 컴포넌트가 렌더링된 이후 또는 특정 상태/props 변경 시 부수적인 작업(예: 데이터 가져오기, 이벤트 리스너 등록 등)을 수행합니다.       |
| useState       | React의 Hook으로, 함수형 컴포넌트에서 상태(state)를 관리합니다.   |
| searchParams   | React Router의 기능으로, URL의 쿼리 파라미터를 관리합니다.  |
| params         | React Router에서 라우터의 URL 매개변수에 접근하는 객체입니다.   |
| useContext     | React의 Hook으로, Context API를 활용하여 계층 구조 깊은 컴포넌트에서 props 없이 데이터(상태나 함수)를 공유하는데 사용됩니다.               |

## 9. References

- Describe all of your references (book, paper, technical report etc).  
12pt, 160%.

[1] HTTP 상태 코드

<https://developer.mozilla.org/ko/docs/Web/HTTP/Status>

[2] state diagram 참고

<https://drawio-app.com/blog/uml-state-diagrams-with-draw-io/>