

# PNU SW 학습공동체 최종(중간)보고서

## Github 로 파일업로드하여 제출

### 1. 프로젝트 소개

프로젝트명: BookUs

가. 배경 및 필요성

올해 초부터 동기들과 각자 책을 한 권씩 정하고 서로 책을 교환하여 읽는 '독서 클럽' 활동을 진행하고 있던 와중 책을 교환하는 과정에서 어려움이 있었음. 교환은 주로 무작위 추첨을 통해 결정되었는데 이 과정에서 이미 지난번에 읽었던 책이 배정되는 사람이 있어 불편함을 느낌. 그래서 각자 이미 교환해서 읽은 책을 제외하고 랜덤으로 다음 책을 정해주는 기능이 있으면 좋겠다고 생각함. 또한 독서 후 감상 기록이 개인별로 기록되거나 서로 공유되지 않아 어플을 통해 독서그룹을 생성하여 비대면으로 책을 서로 교환하여 이 책에 대한 한줄평 및 인상 깊은 내용을 한 번에 모아서 볼 수 있는 기능 또한 추가되면 좋다고 생각함. 각각 개인마다 읽은 독서 이력을 분석하고 감상 기록에 대한 데이터를 수집함. 이러한 분석으로 앞선 문제점을 해결해 나갈 수 있음.

나. 개발목표 및 주요내용, 세부내용 등 개발목표

및 주요내용:

본 프로젝트는 독서 모임 활동을 효율적으로 운영하기 위한 모바일 어플리케이션을 개발하는 것을 목표로 한다. 최근 독서가 트렌드로 자리 잡으면서 다양한 형태의 독서 모임이 활성화되고 있다. 실제로 올해 초부터 동기들과 각자 책을 한 권씩 골라 돌려 읽는 모임을 운영하면서 여러 불편함이 존재했다. 예를 들어, 매주 책을 교환할 때 무작위로 다음 책을 정했는데, 종종 이미 지난 교환 시간에 읽었던 책이 다시 배정되는 문제가 반복되었고 각자 어떤 부분이 인상 깊었는지, 책을 읽고 어떤 생각이 들었는지 공유하기도 불편했다. 이러한 불편함을 해결하고 더 나은 독서 경험을 만들기 위해 본 프로젝트를 기획하게 되었다.

본 프로젝트는 '그룹 기능'에 초점을 맞춘 독서 모임 관리 앱이다. 단순히 개인 독서 활동을 기록하는 것을 넘어, 여러 명이 함께 책을 읽고 공유하는 과정에서 발생하는 다양한 상황을 지원하는 데 중점을 두었다. 특히 실제 독서 모임 경험에서 겪었던 불편함을 해소하고자 구체적인 기능들을 설계했다.

앱은 사용자가 직접 독서 그룹을 만들고 운영할 수 있는 기능을 중심으로 구성되어 있다. 그룹 생성 시에는 대면/비대면 여부, 모임 일정, 그룹 이름/사진/인원 제한 등을 설정할 수 있으며, 필요 시 방장의 승인이 필요한 가입 절차를 통해 그룹 관리를 할 수 있도록 한다.

독서 방식은 두 가지로 나뉜다. 하나는 '한 권의 책을 다 같이 읽기'로, 다함께 책 한 권을 골라 등록하여 모든 그룹원이 정해진 기간 내에 함께 읽는다. 이때 자신의 읽은 쪽수를 기록할 수 있으며, 각자 어느정도 읽었는지 기록하여 모두의 진행 상황을 공유할 수 있다. 또한 인상 깊은 문장, 메모, 한줄평 등을 남기고 이를 다른 구성원들과 공유할 수 있다.

다른 하나는 '각자가 선정한 책을 돌려 읽기'이다. 구성원 각자가 한 권씩 책을 등록하고, 일정 주기에 따라 책을 교환해가며 읽는다. 실제 독서 모임 활동 중에 이미 읽은 책이 다시 배정됐던 문제를 해결하기 위해, 개개인이 이미 읽은 책을 제외한 상태에서 새로운 책을 무작위로 배정해주는 기능을 구현하고자 한다. 아울러 해당 책을 다 읽은 구성원만 책에 대한 다른 사람들의 기록을 열람할 수 있도록 제한하여 읽지 않은 책에 대한 정보 노출을 최소화한다.

그 외에도 마이페이지를 통해 읽은 책 리스트와 감상 기록을 개인별로 확인할 수 있다. 전반적으로 이 앱은 실질적인 독서 모임 운영 경험을 바탕으로 설계되었으며 실제 사용자들의 활동 흐름을 고려한 실용적인 기능으로 구성되어 있다.

세부내용 :

## -요구사항 분석

1. 책 교환 과정에서의 무작위 추천
2. 일정 관리

## -제한사항 및 대책

1. 도서 데이터 수집 및 상세정보 불러오기    1-1. 크롤링 또는 API 를 이용하여 해결
2. 서버 보안
  - 2-1. 로그인 인증 수단 추가 및 비밀번호 해싱
  - 2-2. 로큰 리프레쉬
  - 2-3. CORS 를 허용하여 프론트엔드와 통신 보안

## -개발환경

1. Visual Studio Code

사회적가치 도입 계획 도서모임 어플리케이션 개발을 통해 기존 도서모임의 문제점(책교환 과정에서의 혼선, 일정관리의 어려움 등)을 해결하며, 보다 개선된 도서 모임을 통한 범사회적 도서 증진 가능성.

도서 모임 어플리케이션 사업화를 통한 구글플레이스토어 및 APP Store 와 같은 모바일 어플리케이션 유통 플랫폼 출시 가능성. 향후 사업화 진행 시 책 유통 서비스를 통하여 보다 편리한 책 교환을 통한 어플리케이션 발전 가능성. 추후 모바일 뿐만이 아닌 PC 환경에서 접속 가능한 웹페이지 개발.

추후 정보컴퓨터공학부 학생들을 대상으로 베타 서비스 배포 후 설문조사를 통한 사용자 맞춤화 및 성능 서비스 개선 예정

## 2. 상세설계



- 그룹 내 공지사항
  - .. 그룹 내에서 작성된 공지사항 목록 및 미리보기 확인
  - ! 각 공지사항 우측의 삼각형 아이콘 클릭 시, 공지사항 전체 글 확인 가능
  - ! 공지사항 상단의 펜 모양 아이콘 클릭 시, 공지사항 작성 가능



#### 그룹 내 책바퀴

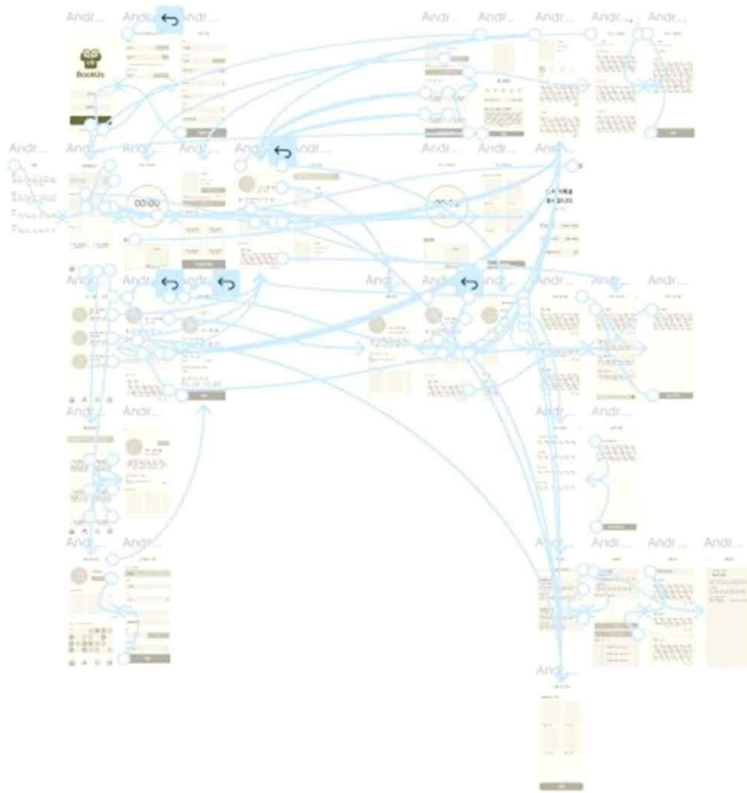
책바퀴 기능을 통해 그룹의 멤버들은 동일한 책을 함께 읽으며 감상을 공유하거나, 각자의 책을 서로 돌려 읽으며 여러 권의 책에 대한 감상을 공유할 수 있음

1. 책바퀴 생성 시, 읽을 책의 권 수를 선택하며, 기간을 설정해두고 정기적인 독서 모임을 가질 수 있음.
2. 책바퀴 홈에서는 이름, 소개글, 참여중인 멤버 등 기본 정보 확인 및 책 교환 및 독서 기간 기록 확인 가능
3. '책 교환하기' 버튼 클릭 시, 여러 권의 책을 나눠 읽을 때 책을 교환하기에 용이한 기록 및 교환 가이드 제공
4. '책바퀴 게시판' 버튼 클릭 시, 책바퀴 내의 게시판으로 이동하며, 해당 게시판에서 사용자들이 작성한 게시글 확인 가능



#### 그룹 책 연동

본인이 현재 읽고 있는 책 중 그룹에 기록을 연동할 책 선택 (연동 시, 해당 책의 독서 기록장에 기록된 글 공유 가능)



# 1) App 전체 구상도



2) Page별 상세 디자인 및 기능 소개

2-1) 로그인 및 회원가입



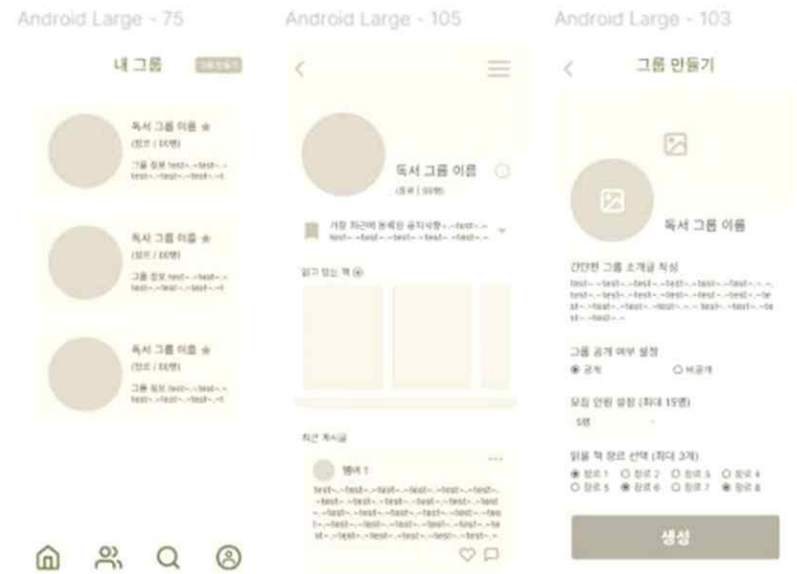
2-2) APP 홈 화면 및 연결 화면



APP 홈 화면 기능

1. 독서 타이머 (독서 시간 기록 후 현재 읽고 있는 책을 읽은 총 누적 시간 저장)
2. 지금 읽고 있는 책 확인 및 화면 전환 (책 클릭 시, 읽고 있는 책 화면으로 전환)
3. 즐겨찾는 그룹 확인 및 그룹 화면 전환 (즐겨찾는 그룹의 홈으로 이동)

2-3) 내 그룹 화면 및 연결 화면



언더바의 내 그룹 아이콘(두 번째 아이콘) 클릭 시, 내 그룹 화면으로 이동

1. 사용자가 참여하고 있는 그룹 목록 확인 및 각 그룹 홈으로 이동 가능
2. 내 그룹 화면 상단의 '그룹 만들기' 버튼 클릭 시, 본인이 그룹장인 독서 그룹을 생성 가능 [독서 그룹 생성 시, 설정 정보: 그룹 이름 및 소개글, 그룹 공개 여부, 모집 인원, 읽을 책 장 르 등]

## 2-4) 그룹 둘러보기 화면 및 연결 화면

Android Large - 77



Android Large - 86



언더바의 둘러보기 아이콘(세 번째 아이콘) 클릭 시, 그룹 둘러보기 화면으로 이동

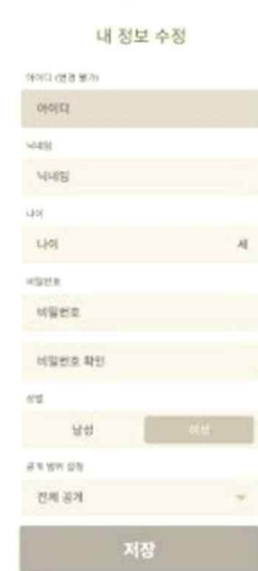
1. 최근에 생성된 독서 그룹 목록 확인 및 각 그룹 미리보기 화면으로 이동 가능
2. 아직 참여하지 않은 독서 그룹의 미리보기 화면의 경우, 독서 그룹 이름, 장르, 인원, 소개글, 그룹 멤버 정보, 현재 읽고 있는 책 등의 일부 정보만 확인 가능
3. 독서 그룹 미리보기 화면 상단의 '참여하기' 버튼 클릭 시, 해당 그룹에 참여 신청을 보낼 수 있으며, 그룹장의 승인 후 독서 그룹에 참여 가능

## 2-5) 마이 페이지 화면 및 연결 화면

Android Large - 76



Android Large - 81



언더바의 마이 페이지 아이콘(네 번째 아이콘) 클릭 시, 마이 페이지 화면으로 이동

1. '내 정보 수정' 버튼 클릭 시, 회원 가입 시 입력한 정보 수정 가능 (아이디 제외)
2. 지금까지 읽은 책(현재 읽고 있는 책 제외) 목록 확인 가능
3. 각 달 별 책을 읽은 페이지 수를 기준으로 달력에 색으로 표시 (읽은 페이지 수가 많을수록 달력에 해당 일자가 진한 색으로 표시됨)

## 2-5) 읽고 있는 책 화면 및 연결 화면



### 읽고 있는 책 화면 기능

1. '독서 시작하기' 버튼 클릭 시, 독서 타이머 화면으로 이동
2. '독서 기록장' 버튼 클릭 시, 해당 책 독서 기록장 화면으로 이동
3. 해당 책을 등록한 내 그룹 확인 가능
4. '다 읽었어요!' 버튼 클릭 시, 책 별점 및 한줄 감상 작성 화면으로 이동 및 다 읽은 책으로 처리

## 2-6) 독서타이머 화면 및 연결 화면



### 독서 타이머 화면 기능

1. 독서 시작 및 종료 시, 독서 시간 기록 가능
2. '책 변경' 버튼 선택 시, 현재 읽고 있는 책 목록 및 선택 화면을 통해 읽을 책 선택 가능 (책 변경은 타이머 시작 전에만 가능)
3. 독서 기록 종료 시, 독서 기록 종료 화면 이동 (진도율 기록, 책 누적 독서 시간, 독서 기록장 이동)



## 2-7) 그룹 홈 화면

Android Large - 104



Android Large - 107



Android Large - 88



### 내 독서 그룹 홈 (두 번째 화면)

1. 독서 그룹 이름 옆 'i' 버튼 클릭 시, 그룹 정보(소개글, 멤버 정보, 읽고 있는 책) 확인 가능
  2. 상단의 메뉴 버튼 클릭 시, 사이드 바 표시
- 아래 화면은 그룹 홈의 메뉴 사이드 바에서 해당 메뉴를 클릭했을 때, 이동하는 화면임.

Android Large - 92



Android Large - 100



Android Large - 101



### 그룹 내 자유 게시판

1. 멤버들이 작성한 게시글 목록 및 미리보기 확인
2. 각 게시글 클릭 시, 좋아요/댓글 등의 상호작용 가능
3. 자유 게시판 상단의 펜 모양 아이콘 클릭 시, 게시글 작성 화면으로 이동 및 게시 가능
4. 각 게시물의 고정, 수정, 삭제 가능

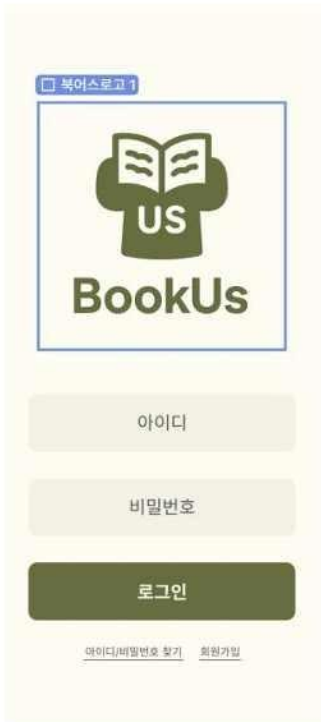
-사용기술

프론트엔드 : Flutter 사용 백엔드 : FastAPI 사용,  
MySQL 사용, 알라딘 API 사용

### 3. 개발결과

-프론트엔드 진행상황

#### <로그인 화면>



해당 화면을 어플리케이션 접속 시 가장 처음 실행되는 화면으로 중앙에는 어플리케이션의 이름인 '북어스'의 로고를 배치시켰다.

로고의 하단에는 아이디와 비밀번호를 입력 할 수 있는 칸이 있으며 비밀번호 입력칸은 개인정보 보호를 이유로 실제 입력값 대신 검은색 동그라미로 대체하여 구현했다. 아이디와 비밀번호 입력칸 하단에는 로그인 버튼이 있으며, 아이디와 비밀번호가 일치할 시 로그인 버튼을 눌러 홈화면에 접속할 수 있다. 아이디와 비밀번호 일치 로직은 추후 백엔드와 연결 후 보완 예정이다.

## <아이디/비밀번호 찾기 및 회원가입>

<

회원가입

닉네임

닉네임

email 인증

email

인증코드 전송

인증 코드

확인

나이

나이

세

아이디

아이디

비밀번호

비밀번호

비밀번호 확인

비밀번호 확인

가입하기

로그인 버튼 하단에는 '아이디/비밀번호 찾기', '회원가입' 칸이 있다. '아이디/비밀번호 찾기' 칸을 누를 시 아이디와 비밀번호를 찾을 수 있는 화면으로 연결되며 추후 보완 예정이다. '회원가입' 칸을 누를 시 회원가입을 할 수 있는 양식이 있는 화면으로 연결되며, 회원가입 양식은 다음과 같다. 닉네임, email, email 인증 코드, 나이, 아이디, 비밀번호, 비밀번호 재확인이 있으며, 올바른 양식 입력 시 회원 가입 버튼을 눌러 회원가입이 가능하다. 보완 설명을 보태자면, email 입력 후 우측에 있는 '인증코드 전송' 버튼을 누르면 해당 이메일로 랜덤 인증코드가 전송되며, 인증코드란에 입력 후 '확인' 버튼을 눌러 일치 시에만 가입이 가능하다. 회원가입 화면은 추후 백엔드와 연결하여 보완 할 예정이다.



### 홈화면 (HomeScreen.dart)

- 작은 타이머
  - 타이머 박스를 누르면 타이머 화면(timer.dart)으로 들어간다. 재생 버튼을 누르면 타이머가 시작된다.
- 지금 읽고 있는 책 리스트
  - 글자 옆 플러스 버튼을 누르면 지금 읽고 있는 책의 전체 리스트 화면으로 이동한다.
  - 책 표지를 행스크롤로 최대 5개까지 보여줄 수 있다.
- 즐거찾는 그룹 리스트
  - 즐겨찾는 그룹에 편하게 접근할 수 있도록 홈화면에 배치했다.
  - 그룹 사진, 이름, 장르, 인원 수를 확인할 수 있다.
  - 행스크롤로 배치하였다.



### 타이머 큰화면 (timer.dart)

- 원형 타이머
  - 재생 버튼을 누르면 타이머가 시작되고, 원 테두리의 작은 침이 한 시간에 한 바퀴씩 회전하도록 했다.
  - 종료 버튼을 누르면 "독서 기록을 종료하시겠습니까? (취소 / 종료하기)"라는 질문이 뜨고, 취소 버튼을 누를 시 계속해서 타이머가 진행되고 종료하기 버튼을 누르면 독서 기록 종료 페이지(reading\_complete\_screen.dart) 화면으로 이동한다.
- 현재 읽고 있는 책
  - 현재 읽고 있는 책의 표지, 제목, 작가, 페이지를 보여준다.
  - 사용자가 입력했던 마지막으로 읽었던 페이지를 이용하여 진도율을 비율로 보여준다.



### 3. 책 변경

- 책 변경 버튼을 누르면 현재 읽고 있는 책 목록이 뜨고, 이들 중 하나를 선택해(라디오버튼) 읽는 책을 변경할 수 있도록 구현했다.



### 독서 기록 종료 페이지 (reading\_complete\_screen.dart)

- 사용자가 진도율을 입력하여 현재 읽은 페이지를 업데이트 할 수 있다.
- 읽은 책의 누적 독서 시간을 보여준다. 직전에 했던 타이머의 시간이 아닌, 현재 책의 이때까지 읽었던 누적 시간을 보여준다.
- 독서 메모 남기기의 메모 아이콘을 누르면 인상 깊은 구절/메모를 남길 수 있는 페이지로 이동한다. (메모 페이지 구현 예정)



### 그룹 검색 페이지(group\_search\_screen.dart)

- 현재 검색창까지 구현하였다.
- 2자 이상, 20자 이하만 검색 가능하도록 설정하였다.

## -백엔드 진행상황

**POST** /users/health/health

Parameters

Request body

```
{
  "url": "profibonus.ac.kr"
}
```

Responses

200

```
{
  "msg": "성공해 200 명분식명입니다."
}
```

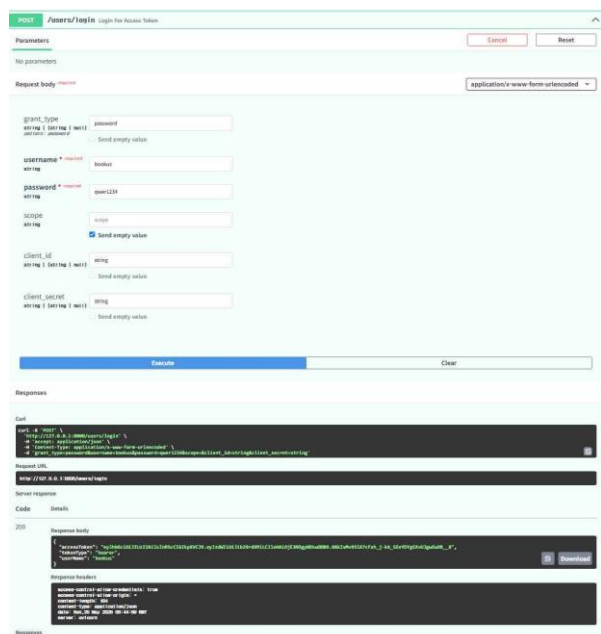
Download



회원이입시 이메일 인증을 위해 인증코드를 메일로 전송하도록 구현

[illegible]

회원가입시 데이터 베이스에 유저 정보가 들어감을 확인할 수 있음. 비밀번호는



클라이언트가 /books/search/{keyword} get요청을 보내면 알라딘 api list검색을 사용하여 반환된 값을 json형식으로 파싱하여 클라이언트에 전송한다



클라이언트에서 /books/add/{isbn13} post요청을 하면 isbn을 이용해 알라딘 api상세검색을 하고 반환된 값을 파싱하여 책 이름, 저자, 페이지 수, isbn, 책저장을 요청한 userName 등등의 정보를 books table에 저장한다



CryptContext 라이브러리를 사용하여 해싱하여 저장함. 이후 로그인 시 토큰을

발급하도록 구현, 토큰을 이용하여 현재 사용자 정보를 불러오는  
get\_current\_user 함수 구현.



**POST** /groups/createGroup Create Group

Parameters

token \* required eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ...

Request body \* required application/json

```
{
  "groupName": "string",
  "groupLeader": "string",
  "groupExpire": "string",
  "openOrClose": 0,
  "password": 0,
  "memberLimit": 0
}
```

Result Grid

	id	groupLeaderName	groupName	groupExpire	groupGenre	openOrClose	password	memberLimit
3	hobin	string	string	string	0	0	0	

groups 1 x

로그인을 통해 토큰을 발급받고 그 토큰을 이용하여 그룹을 생성함

**DELETE** /groups/deleteGroup/{groupId} Delete Group

Parameters

groupId \* required 3

token \* required eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ...

Execute Clear

Responses

Curl

```
curl -X 'DELETE' \
  'http://127.0.0.1:3000/groups/deleteGroup/{groupId}' \
  -H 'accept: application/json'
```

Request URL

Server response

Code Details

200

Response body

```
{
  "delete completely"
}
```

Response headers

```
Access-Control-Allow-Origin: http://localhost:3000
Access-Control-Allow-Headers: origin, x-requested-with
Access-Control-Allow-Methods: GET, POST, PUT, DELETE, OPTIONS
Content-Type: application/json
Date: Mon, 10 Oct 2017 10:00:00 GMT
Server: Express
```

그룹 삭제 권한은 그룹을 생성한 그룹장만 있기에 파라미터로 그룹아이디와 로그인 시 발급받은 토큰을 보낸다. 그룹아이디와 그룹리더 이름이 일치한다면 그룹을 삭제한다.



## 개발결과

### users



POST	/users/login	Login For Access Token	▼
POST	/users/signup	Signup	▼
POST	/users/email/send	Sendverifyemail	▼
POST	/users/email/verify	Verify Code	▼
PUT	/users/me	Update Me	▼
GET	/users/profile/{userName}	Get User Profile	▼
POST	/users/password-reset/request	Request Password Reset	▼
POST	/users/password-reset/complete	Reset Password	▼
POST	/users/find-id/request	Request Find Id	▼
POST	/users/find-id/verify	Verify Find Id	▼
GET	/users/myprofile	Get My Profile	▼
POST	/users/usrImage	Usr Image	▼
PUT	/users/updateImage	Update Img	▼

users 라우터로 로그인, 회원가입, 이메일 인증이 가능하게 만들어 줬고, 자신의 정보를 반환하거나, 다른사람의 정보를 반환하는 것도 가능하게 만들었다. 또한 자신의 정보를 수정하는 기능도 넣어 주었다.

groups			^
POST	/groups/createGroup	Create Group	▼
GET	/groups/allGroup	Show All Group	▼
GET	/groups/myGroup	Show My Group	▼
PUT	/groups/updateGroup/{groupId}	Update Group	▼
DELETE	/groups/deleteGroup/{groupId}	Delete Group	▼
POST	/groups/joinGroup/{groupId}	Join Group	▼
DELETE	/groups/leaveGroup/{groupId}	Leave Group	▼
GET	/groups/sameBookGroup/{isbn}	Same Book Group	▼
GET	/groups/groupInfo/{groupId}	Group Info	▼
GET	/groups/myGroupInfo/{groupId}	Mygroup Info	▼
GET	/groups/isLeader/{groupId}	Is Leader	▼
POST	/groups/image	Image	▼

Post router를 통해 그룹 생성, 그리고 해당 그룹의 id를 통해 그룹 가입, 그룹 프로필 및 배경 이미지를 추가할 수 있는 기능을 구현했다. 그룹 가입의 경우는 공개/비공개에 따라 비밀번호 입력 여부도 추가하였다. 그룹이미지는 프론트에서 받아온 사진원본을 서버에 저장하고 그 사진의 경로를 데이터베이스의 저장하는 식으로 구현하였다. Get을 통해 토큰에 따라 모든 그룹의 대한 정보와 내 그룹의 대한 정보를 얻을 수 있다. 이때 그룹 정보 같은 경우 해당 그룹이 공개/비공개에 따라서 사용자가 얻을 수 있는 정보의 양이 다르다. 예를 들어 공개라면 해당 그룹의 모든 정보를 볼 수 있지만 비공개라면 그룹이름, 그룹이미지, 최대인원 등 얻을 수 있는 정보의 제한이 있다. 책의 isbn을 통해 해당 그룹 내에서 그룹책과 같은 책이 등록되어 있는 그룹도 얻을 수 있다. 그룹 리더를 판단하는 라우터를 통하여 그룹리더, 그룹원이라는 정보에 따라 delete router를 통해 그룹 삭제 / 떠나기를 할 수 있도록 기능을 구현하였다. 삭제의 경우 그룹장이 그룹을 삭제하면 현재 데이터베이스의 남아있는 그룹의 대한 모든 정보가 삭제되고 떠나기를 하면 해당 그룹원의 대한 정보만 삭제된다. Post router를 통하여 그룹의 최대인원, 장르 등을 수정할 수 있도록 하였다.

favoritegroups			^
POST	/groups/favorite	Post Favorite	▼
GET	/groups/favorites	Get Favorites	▼

내 그룹 중에서도 좋아하는 그룹을 따로 설정할 수 있도록 구현하여 내 그룹 내에서도 차별화를 두었다.

#### notifications\_router

POST /notifications/createNotice/{groupId} Create Notice

PUT /notifications/updateNotice/{groupId}/{id} Update Notice

DELETE /notifications/deleteNotice/{groupId}/{id} Delete Notice

GET /notifications/showNotice/{groupId} Show Notice

공지사항의 대한 라우터인데 해당 그룹에서 그룹장만이 공지사항을 작성, 수정, 삭제할 수 있다.

#### groupposts\_router

POST /groupposts/postCreate/{groupId} Post Create

PUT /groupposts/updatePost/{groupId}/{id} Update Post

DELETE /groupposts/deletePost/{groupId}/{id} Delete Post

GET /groupposts/showPost/{groupId} Show Post

그룹 내에서 사람들과 다양한 의견을 공유할 수 있도록 그룹게시판을 만들었다. 여기서 그룹게시글 같은 경우는 위의 공지사항과 달리 사용자가 해당 그룹원이라면 누구든지 그룹게시글을 등록할 수 있게 하였다.

#### comments\_router

POST /comments/writeComment/{groupId}/{id} Write Comment

PUT /comments/updateComment/{groupId}/{postId}/{id} Update Comment

DELETE /comments/deleteComment/{groupId}/{postId}/{id} Delete Comment

GET /comments/showComment/{groupId}/{postId} Show Comment

댓글 기능도 마찬가지로 사용자가 해당 그룹원이라면 누구든지 특정 게시글에 대하여 댓글을 작성할 수 있다.

#### hearts\_router

POST /hearts/toggleHeart/{groupId}/{postId} Toggle Heart

GET /hearts/showHeart/{groupId}/{postId} Show Heart

좋아요 기능도 마찬가지로 구현하였다.

books			^
GET	/books/search/{keyword}	Search Books	▼
POST	/books/add/{isbn13}	Add Book	▼
DELETE	/books/delete/{isbn13}	Delete Book	▼
GET	/books/bookInfo/{isbn}	Get Book Info	▼
GET	/books/info/{isbn13}	Get Book Info	▼
GET	/books/infolist/{option}	Get Books List	▼
PATCH	/books/startrecord/{isbn13}	Endrecord	▼
PATCH	/books/endrecord	Endrecord	▼
PATCH	/books/endread	Endread	▼

Books 데이터베이스에는 다양한 router가 있는데

/books/search/{keyword} keyword로 알라딘 api를 호출하고 books list를 프론트에 반환해준다

/books/add/{isbn13}은 책 등록번호인 isbn13을 이용해 database에 책 정보를 저장한다

/books/delete/{isbn13}을 이용해 해당 유저의 해당 isbn 즉 책 정보를 데이터베이스에서 삭제한다

/books/bookInfo/{isbn} 을 이용해 책 정보(저자, 책 페이지)를 조회가능하다

/books/info/{isbn13} 을 통해서 해당 유저가 등록한 책 정보(읽기 정보)를 조회가능하다

/books/infolist/{option}을 통해서 option이 0이면 읽는중인 책만 1이면 완독한 책만 2면 완독중이거나 읽는중이거나 상관없이 해당 사용자의 데이터베이스에 있는 책을 모두 반환한다

/books/startrecord/{isbn13}을 통해서 책 접근시간을 업데이트하여 프론트 화면에서 정렬하여 보여지도록 한다

/books/endrecord 를 통해서 읽기를 중단하고 진도율, 독서시간 등을 데이터베이스에 patch한다

/books/endread를 통해서 한 책에 대한 독서상태를 완독으로 바꾸고 별점, 한줄평 등을 데이터베이스에 저장한다.

# bookmemo



POST	/bookmemo/post	Post Book Memo	▼
DELETE	/bookmemo/delete/{memo_id}	Delete Book Memo	▼
PATCH	/bookmemo/modify	Modify Book Memo	▼
PATCH	/bookmemo/marking	Marking	▼
GET	/bookmemo/infolist/{textType}	List Book Memos	▼
GET	/bookmemo/member/infolist	Member Infolist	▼

/bookmemo/post를 사용해서 책에 대한 인상깊은 구절, 메모 등을 구별하여 저장한다

## bookmemo



POST /bookmemo/post Post Book Memo

Parameters Try it out

Name	Description
token <small>required</small>	token
string (query)	

Request body required

application/json

Example Value Schema

```
{  "bookIsbn": "string",  "textType": 0,  "content": "string",  "openRange": 0,  "contentPage": 0}
```

메모는 textType 0 인상깊은 구절은 1로 저장한다

메모, 인상깊은 구절마다 공개하고 싶은 범위는 openRange를 설정하여 0이면 비공개 1이면 친한그룹 공개 2면 전체공개로 설정하고 contentPage에는 어떤 페이지에 대한 메모, 인상깊은 구절을 작성하는 지 저장한다

/bookmemo/delete/{memo\_id}를 통해 메모의 primary\_key인 id로 데이터베이스에서 삭제한다 그리고 삭제 과정에서 사용자확인도 하여 데이터손실을 막아준다

/bookmemo/modify를 통해서 책메모의 공개범위, 내용, 페이지 전부 수정할수있다

/bookmemo/marking을 통해서 책메모에 즐겨찾기를 할수있다 즐겨찾기는 메모, 인상깊은구절마다 3개이며 이 즐겨찾기는 앱화면에서 상단에 위치한다

/bookmemo/infolist/{textType}을 통해 메모 또는 인상깊은 구절을 앱사용자에게 반환해준다

/bookmemo/member/infolist를 통해 그룹에서 책에대한 메모정보를 확인할 때 그룹멤버가 설정해놓은 공개범위를 토대로 사용자에게 적절하게 메모를 필터링하여 반환하고 앱 화면에 보여주는 역할을 함

## groupbooks\_router



POST	/groupbooks/post	Add Group Book	✓
DELETE	/groupbooks/delete	Delete Group Book	✓
GET	/groupbooks/list/{groupId}	Get Group Books List	✓
GET	/groupbooks/readers	Get Book Readers	✓

/groupbooks/post를 통해 그룹에 책을 등록한다

/groupbooks/delete를 통해 그룹에 등록된 책을 삭제한다

/groupbooks/list/{groupId}를 통해 primaryKey인 id를 받고 해당 그룹에 등록된 책을 모두 반환한다

/groupbooks/readers를 통해 해당 그룹에서 어떠한 책을 누가 읽는지 반환해준다

## readingpages\_router



POST	/reading-log/	Post Reading Log	✓
GET	/reading-log/	Get Logs	✓
GET	/reading-log/{userName}	Get Logs Of Other User	✓

readingpages\_router는 책잔디 심기 기능에 대한 라우터로,  
하루에 몇시간 읽었는지에 대한 정보를 데이터베이스에 저장한다.

## bookcircle\_router



POST	/bookcircle/createBookCircle/{groupId}	Create Bookcircle	▼
POST	/bookcircle/joinBookCircle/{groupId}/{bookCircleId}	Join Bookcircle	▼
DELETE	/bookcircle/deleteBookCircle/{groupId}/{bookCircleId}	Delete Bookcircle	▼
GET	/bookcircle/showBookCircle/{groupId}	Show Bookcircle	▼

## bookcirclemember\_router



GET	/bookcirclemember/showBookCircleMember/{groupId}/{bookCircleId}	Show Bookcirclemember	▼
-----	---	-----------------------	---

## bookcirclebooks\_router



POST	/bookcirclebooks/post	Post Bookcircle Book	▼
GET	/bookcirclebooks/list	Get Bookcircle Books	▼

## bookcirclelogs\_router



POST	/bookcircle/start	Start Circle	▼
POST	/bookcircle/exchange	Exchange Circle	▼
GET	/bookcircle/is-leader	Is Circle Leader	▼
POST	/bookcircle/end/{circle_id}	End Book Circle	▼
GET	/bookcircle/logs/{circle_id}	Fetch Logs	▼

앱의 주요기능인 책바퀴에 대한 라우터 들로, 책바퀴 생성, 책바퀴 시작, 책교환 시작, 책바퀴 종료에 대한 기능들을 구현했고, 책바퀴 멤버들에 대한 정보반환, 랜덤 책교환 로직 구현, 책교환 로그 데이터베이스 저장의 기능을 하고있다.



위 화면은 앱의 시작화면인 로그인 화면으로 올바른 아이디와 비밀번호를 입력해 앱에 접속할 수 있다.



위 화면은 아이디/비밀번호 찾기 화면으로 email로 인증코드를 발송하여 인증코드 일치 시, 아이디와 비밀번호를 찾을 수 있다.



회원가입

닉네임

닉네임

email 인증

email

인증 코드 전송

인증 코드

확인

나이

나이

세

아이디

아이디

비밀번호

비밀번호

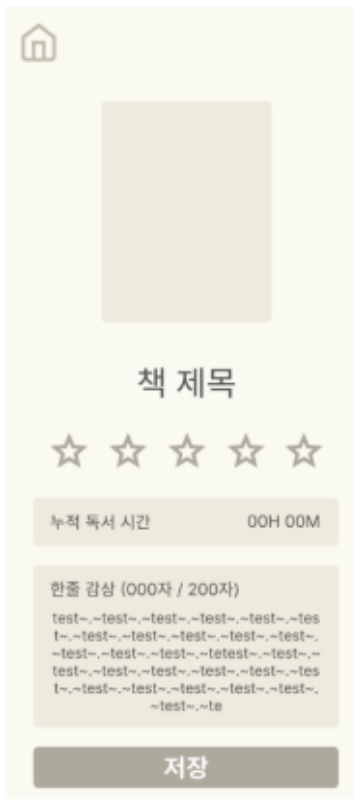
비밀번호 확인

비밀번호 확인

가입하기

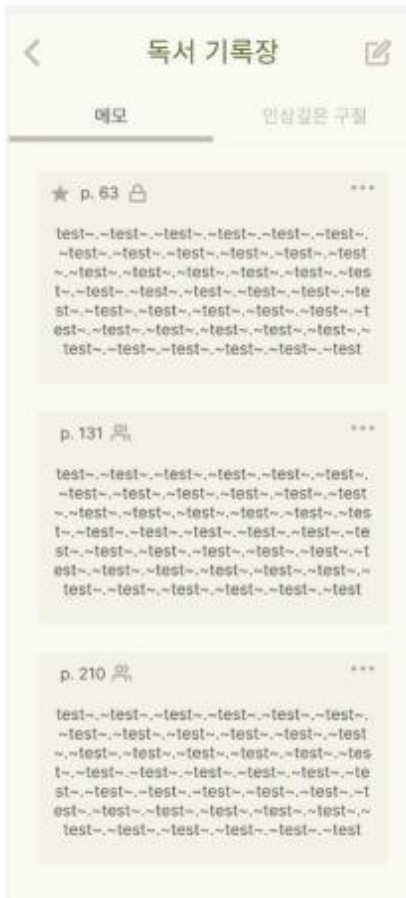


위 화면은 홈화면에서 '지금 읽고 있는 책'을 누르면 연결되는 화면이며, 기본적인 책정보와, 진도율을 보여주며, '독서시작하기' 버튼은 타이머 화면과 '독서기록창 이동' 버튼은 독서기록장 화면과, '다 읽었어요' 버튼은 독서평점 화면과 이어지며, '이 책을 등록한 그룹' 밑 컨테이너를 누를 시 해당 책을 등록한 그룹의 그룹홈과 이어진다.

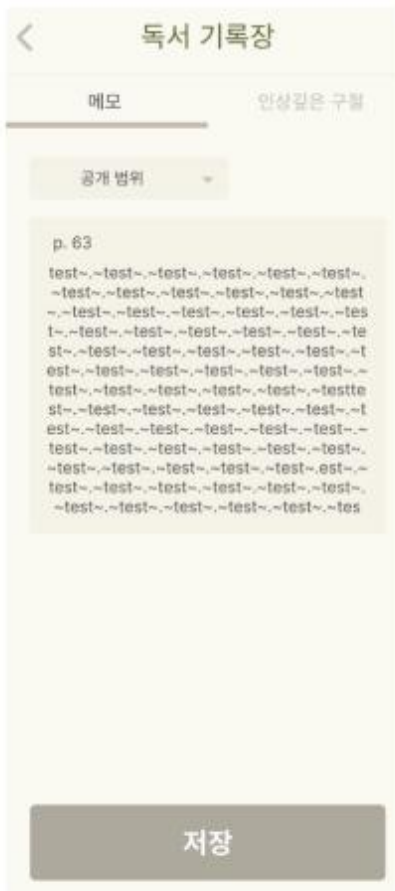


위 화면은 '지금 읽고 있는 책' 화면의 '다 읽었어요' 버튼과 이어지는 화면으로 다 읽은 책의 별점, 누적 독서 시간 및 한줄평 입력이 가능하다.

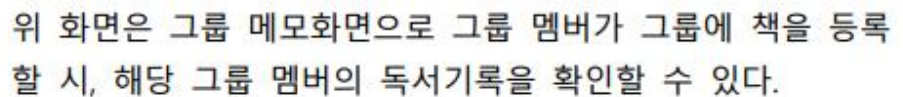
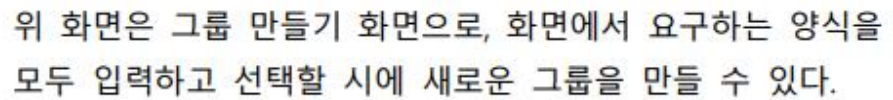




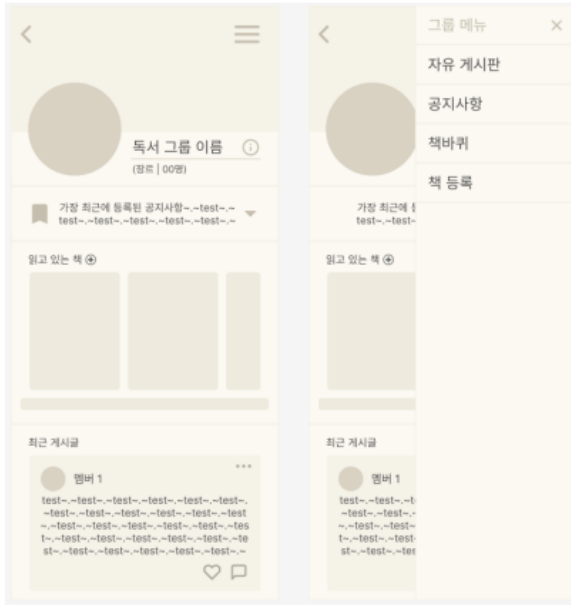
위 화면은 독서기록장 화면으로 해당 책의 메모와 인상 깊은 구절에 대한 독서 기록을 열람할 수 있으며, 페이지 우측의 아이콘을 통해 해당 독서 기록의 공개범위를 직관적으로 파악할 수 있다. 독서 기록 우측 상단의 세개의 점을 눌러 메모의 고정과 수정 및 삭제가 가능하며, 독서 기록 고정 시 페이지 앞 별 아이콘과 함께 가장 상단으로 독서 기록이 이동한다. 우측 상단의 아이콘을 클릭 시 독서 기록 입력 화면으로 이어진다.



위 화면은 독서 기록을 직접 입력하는 화면으로 독서 메모와 인상 깊은 구절을 분리하여 기록할 수 있으며, 공개범위를 유저가 직접 설정할 수 있으며, 페이지와 해당 내용을 입력 후 저장 버튼을 통하여 독서 기록을 저장할 수 있다.



## <그룹 홈 및 메뉴>



각 독서 그룹의 홈 화면 및 메뉴.

독서 그룹의 대표 사진, 배경 사진, 그룹 명, 장르 및 인원 등의 기본 정보가 표시되며, 현재 그룹원들이 읽고 있는 책, 가장 최근 공지사항 및 최근 게시물 3개가 홈 화면에 함께 표시된다.

공지사항 및 게시글의 경우, provider를 이용하여 데이터를 불러온 후 최근 1개 혹은 3개의 데이터를 홈 화면에 표시하는 방식을 이용하였다.

그룹 메뉴를 통해 각 기능을 다루는 페이지로의 이동이 이루어진다.

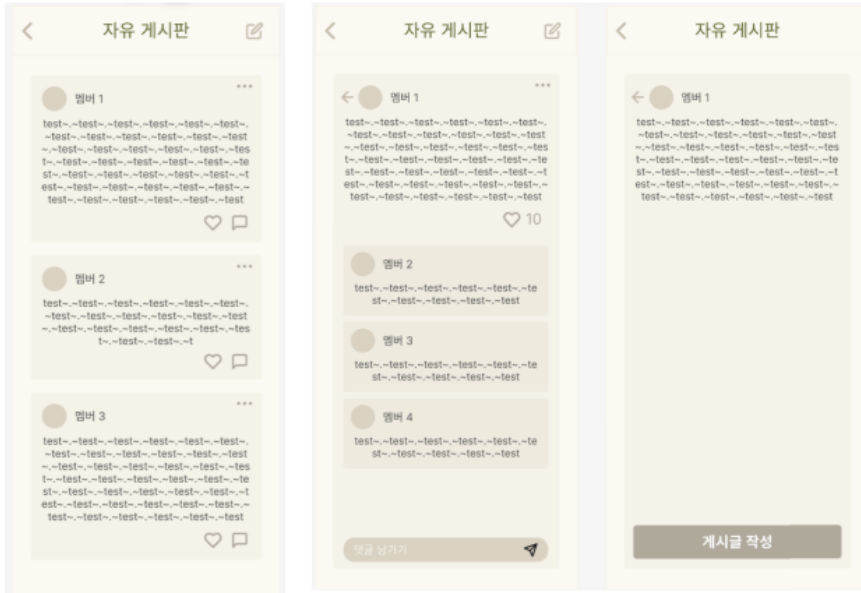
## <그룹 정보>



각 그룹 홈에서 그룹 명 옆의 'i' 아이콘'을 클릭했을 때 이동하는 화면.

그룹 명, 장르, 인원, 그룹 장, 소개 글, 연령 및 성별 정보 등의 기본적인 그룹 관련 정보를 표시한다.

## <그룹 내 자유게시판>



그룹 메뉴에서 '자유게시판' 메뉴를 클릭했을 때 이동하는 화면.

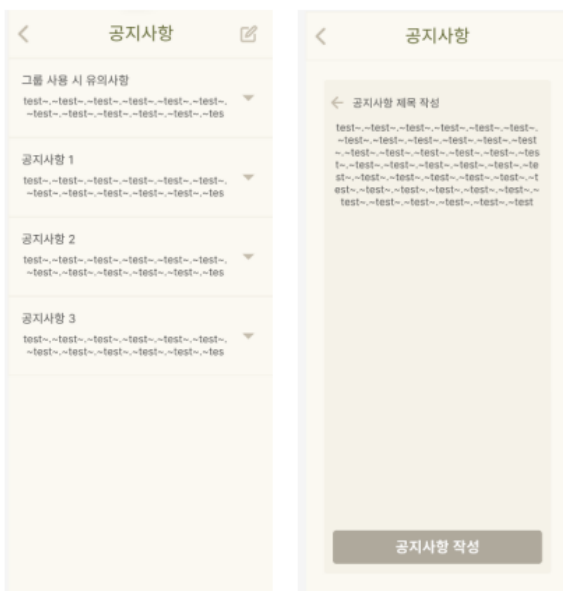
전체 게시글 및 댓글 정보는 class 구조 및 provider를 이용하여 출력하였으며, 멤버 정보, 글 정보, 좋아요 수 등의 데이터를 데이터베이스에서 전달받았다.

좋아요 아이콘 클릭, 게시글 등록, 댓글 등록 등의 변경 사항이 생기는 경우, 데이터

베이스에 새로운 정보를 보낸 후, provider가 받아오는 데이터를 다시 로드 하는 방식으로 화면에 표시되는 정보를 최신화 하였다.

각 게시글 및 댓글에는 수정, 삭제 기능을 부여하였으며, 이러한 기능은 각 게시글 및 댓글의 작성자 정보를 저장한 후, 현재 로그인한 사용자 정보와 비교하여 작성자에게만 해당 기능을 포함한 아이콘이 표시되도록 화면을 구성하였다.

## <그룹 내 공지사항>



그룹 메뉴에서 '공지사항' 메뉴를 클릭했을 때 이동하는 화면.

공지사항은 그룹 장만 작성 가능하며, 이에 따라 각 그룹에서 그룹 장 정보를 불러온 후 로그인한 현재 사용자와 비교하여 작성 버튼 및 공지사항의 수정, 삭제 아이콘을 그룹 장에게만 보이도록 화면을 구성하였다.

그룹 장을 제외한 그룹의 구성원들의 경우, 공지사항에 대한 단순 확인만 가능하다.

## <그룹 책 연동>



그룹 메뉴에서 '책 등록' 메뉴를 클릭했을 때 이동하는 화면.

그룹에 등록하는 책은 현재 사용자가 읽고 있는 개인 책 데이터 중에서 선택하는 경우만 가능하며, 이에 따라 책 연동 페이지 이동 시, 각 사용자의 읽고 있는 책 데이터에 기반한 책 리스트가 표시된다.

그룹에 책을 연동 시, 해당 책에 등록한 개인의 메모, 인상 깊은 구절이 각 메모의 공개 범위에 따라 다른 그룹원들에게 공유된다.





## 홈화면 (HomeScreen.dart)

### 1. 작은 타이머

- 타이머 박스를 누르면 타이머 화면(timer.dart)으로 들어간다. 재생 버튼을 누르면 타이머가 시작되다.

### 2. 지금 읽고 있는 책 리스트

- 글자 옆 플러스 버튼을 누르면 책을 등록하는 화면으로 이동한다.
- 현재 읽고 있는 책을 행스크롤로 보여준다.

### 3. 즐겨찾는 그룹 리스트

- 즐겨찾는 그룹에 편하게 접근할 수 있도록 홈화면에 배치했다.
- 그룹 사진, 이름, 장르, 인원 수를 확인할 수 있다.
- 행스크롤로 배치하였다.



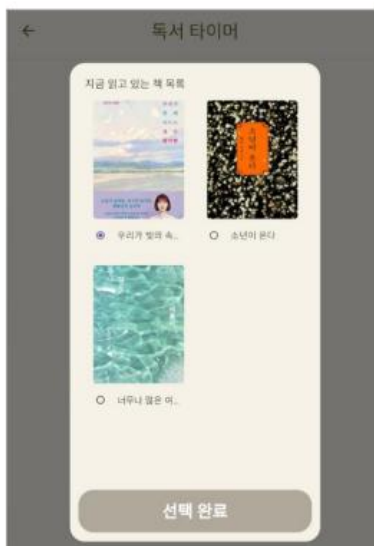
## 타이머 큰화면 (timer.dart)

### 1. 원형 타이머

- 재생 버튼을 누르면 타이머가 시작되고, 원 테두리의 작은 침이 한 시간에 한 바퀴씩 회전하도록 했다.
- 종료 버튼을 누르면 "독서 기록을 종료하시겠습니까? (취소 / 종료하기)"라는 질문이 뜨고, 취소 버튼을 누를 시 계속해서 타이머가 진행되고 종료하기 버튼을 누르면 독서 기록 종료 페이지(reading\_complete\_screen.dart) 화면으로 이동한다.

### 2. 현재 읽고 있는 책

- 현재 읽고 있는 책의 표지, 제목, 작가, 페이지를 보여준다.
- 사용자가 입력했던 마지막으로 읽었던 페이지를 이용하여 진도율을 비율로 보여준다.



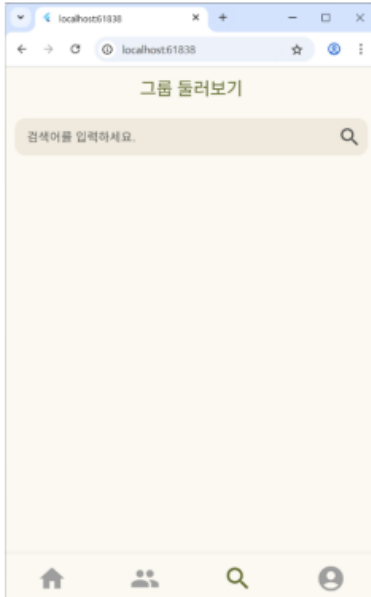
### 3. 책 변경

- 책 변경 버튼을 누르면 현재 읽고 있는 책 목록이 뜨고, 이들 중 하나를 선택해(라디오버튼) 읽는 책을 변경할 수 있도록 구현했다.



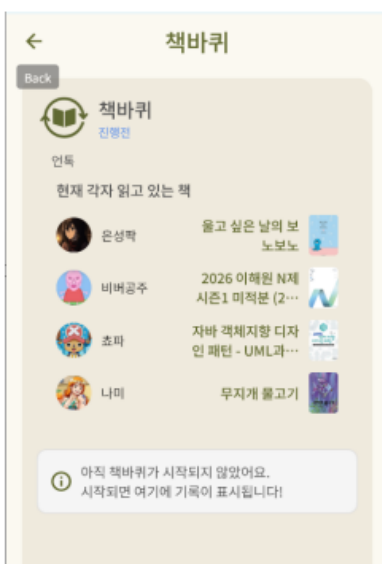
### 독서 기록 종료 페이지 (reading\_complete\_screen.dart)

- 사용자가 진도율을 입력하여 현재 읽은 페이지를 업데이트 할 수 있다.
- 읽은 책의 누적 독서 시간을 보여준다. 직전에 했던 타이머의 시간이 아닌, 현재 책의 이때까지 읽었던 누적 시간을 보여준다.
- 독서 메모 남기기의 메모 아이콘을 누르면 인상 깊은 구절/메모를 남길 수 있는 페이지로 이동한다. (메모 페이지 구현 예정)



### 그룹 검색 페이지(group\_search\_screen.dart)

- 키워드를 입력하면 해당 키워드가 들어간 그룹의 목록을 보여준다.
- 비공개 그룹의 경우 그룹 사진과 이름만 뜨고, 그룹장이 설정한 비밀번호를 입력하여 들어갈 수 있다.
- 공개 그룹의 경우 그룹 사진, 이름, 성별 비율, 평균 나이, 현재 읽고 있는 책 등 많은 정보가 공개되어 있으며 참여하기 버튼을 누르면 바로 해당 그룹에 참가할 수 있다 (시연 영상 참고)



### 책바퀴 기능

- \* 그룹원들이 각자 선택한 책을 서로 돌려 읽는 기능
- 그룹장은 책바퀴 시작하기, 교환하기, 종료하기 버튼이 보이도록
- 그룹원들은 별다른 버튼 없이 책바퀴 히스토리만 보이도록
- 책바퀴 시작 전 아직 참여하지 않은 사람들에게겐 참여하기 버튼이 보이도록 한다.



책바퀴 목록에서 보이는 모습



새 책바퀴를 생성하거나 참여하기 위해서는 반드시 책 한 권을 선택하도록 했다.



## 마이페이지

프로필 사진/이름, 내 정보 수정 버튼, 현재 자신이 읽기를 완료한 책 (읽고 있는 중인 책은 뜨지 않는다.), 이번 달 독서 시간 잔디 심기를 보여주는 마이페이지.

- 오른쪽 상단 버튼을 누르면 로그아웃된다.

← 내 정보 수정

아이디 (변경 불가)  
seoyeonkim

닉네임  
닉네임을 입력하세요

나이  
21 세

비밀번호 변경  
새 비밀번호

새 비밀번호 확인

성별  
남성 여성

공개 범위 설정  
비공개

저장하기

내 정보 수정 버튼을 누르면 보이는 화면  
닉네임, 나이, 비밀번호, 성별, 공개 여부를 수정할 수 있다.

←

사랑과 멸종을 바꿔 읽어보십시오

유선혜 (자은이) | 총 페이지 수

★★★★★

누적 독서 시간 00H 00M

정말 감명깊게 읽었습니다.. so good~

마이페이지에서 지금까지 읽은 책 표지를 누르면 사용자가 예전에 기록했던 별점, 누적 독서 시간, 한줄평을 확인할 수 있다.

시연영상:

<https://www.youtube.com/watch?v=80D73T6l0Ng>

## 6. 팀 소개

팀장: 장호빈(기획, 백엔드)

팀원 1: 곽윤정(디자인, 프론트엔드)

팀원 2: 김기태(백엔드)

팀원 3: 김서연(프론트엔드)

팀원 4: 김희수(프론트엔드)

팀원 5: 박은성(백엔드)