
캡스톤 디자인 최종 보고서

- CAUCLUB -



과목명	캡스톤디자인
교수명	손용석 교수님
제출일	2022.06.10.
학 과	소프트웨어학부
팀 장	20185168 이기태
팀 원	20200604 이예빈
팀 원	20201241 오명진

Table of Contents

I. 개요	4
1) CAUCLUB	4
2) 개발 동기 및 목적.....	4
II. 기존 서비스들과의 비교 / 분석.....	6
1) 공모전 스터디, 대외활동 모집.....	6
2) 랭킹 제도	7
III. 설계	8
1) SWOT 분석.....	8
2) 기능 설계	8
3) 기술 설계	10
IV. 과제 수행 과정 및 구성원 간 역할분담.....	13
1) 개인별 과제 수행 결과	13
2) 팀 스케줄	14
3) 개인별 스케줄	15
V. 기능 상세.....	17
1) 전체 기능에 대한 흐름도.....	17
2) 사용자 관점에서의 어플리케이션 화면.....	18
VI. 기술 특징.....	19
1) 채팅 서비스.....	19
2) 비밀번호 암호화	19
3) 추천 시스템.....	19

VII. 활용방안 및 기대효과.....	20
1) 과동아리 간 소통의 어려움 해결.....	20
2) 과동아리의 전문성 향상.....	20
3) 지역 내 중고등학교 동아리 연합 형성 가능성.....	20
VIII. 향후 발전 방향성.....	20
IX. 결과물.....	22
1) 앱 시작 화면, 회원가입 및 가입 결과.....	22
2) 아이디 찾기 및 실행 결과.....	23
3) 비밀번호 찾기.....	24
4) 분류별 동아리 랭킹.....	25
5) 타과 동아리 프로필.....	26
6) 자과 동아리 프로필 – 가입 / 탈퇴.....	26
7) 동아리 생성.....	27
8) 게시판 글 작성, 수정, 삭제.....	27
9) 아카이브 글 작성, 좋아요, 수정, 삭제.....	28
10) 동아리 추천 -> 마이페이지에서 키워드 수정 -> 재추천.....	29
11) 채팅 : TESTID 유저와 CHAOS 동아리 회장 간의 채팅.....	30
12) 로그아웃, 회원 탈퇴.....	31

[illegible]

위 기사에서는 대학생들이 본인의 취미생활을 중점적으로 두고 동아리를 선택한다는 내용이 포함되어 있다. 이처럼 동아리는 개개인의 전반적인 대학생활과 그 이후의 취업까지, 대학 생활 전반에 영향을 끼치고 있는 중요한 요소다.

우리 최강미디어는 이런 동아리 중에서도 과동아리 활동에서 문제점을 몸소 느낀 팀원들로 구성되어 있다. 과동아리 활동의 불편함을 느꼈고, 이는 CAUCLUB 어플리케이션을 기획하게 된 **직접적 계기**가 된다.

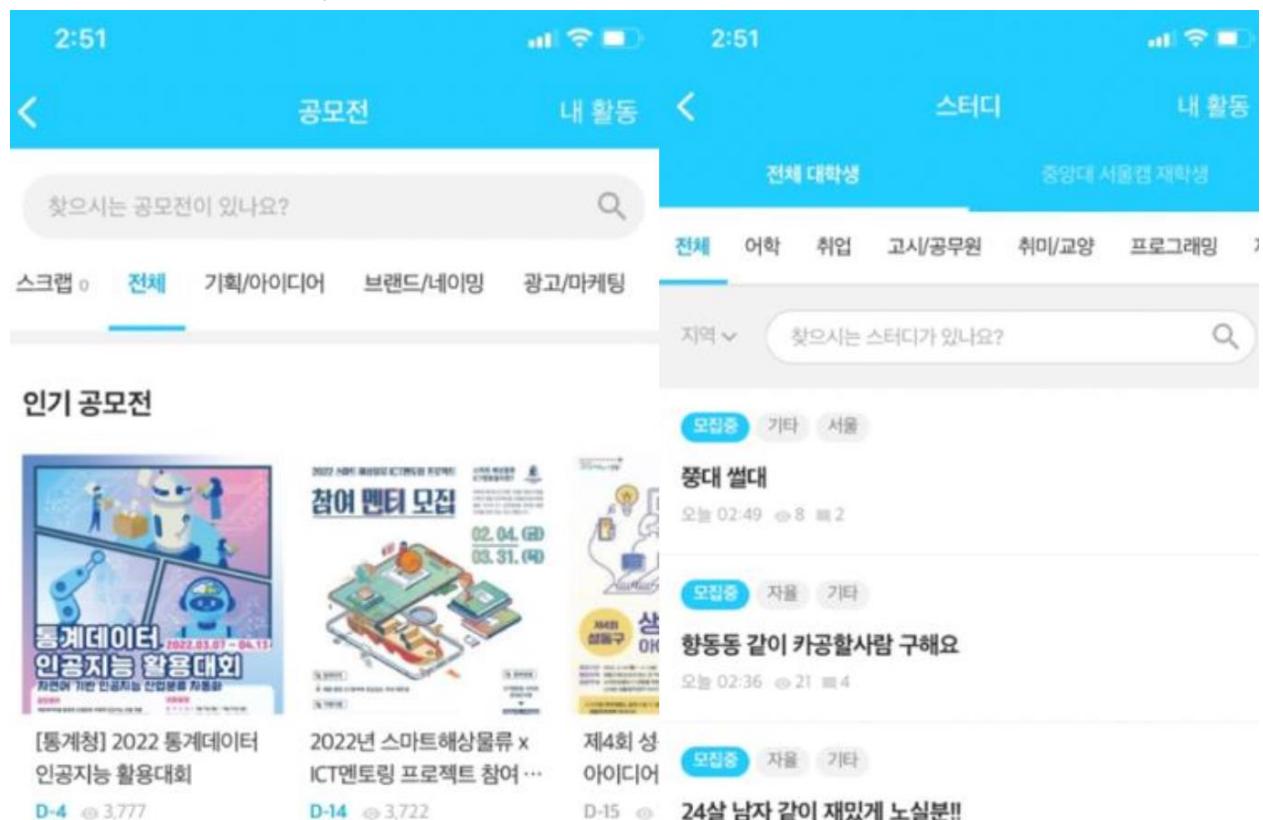
이기태 팀원은 중앙대학교 소프트웨어학부 농구동아리 퍼주마(구 플로터)의 회장을 몇 년째 맡아 오고 있다. 해당 동아리의 가장 큰 문제점은 동아리 인원만으로는 매주 진행되는 게임 운영이 어렵다는 점이었다. 그래서 찾게 된 방법이 다른 학과와의 친선경기였고, **동아리 회장에게 연락하고자 했으나 핸드폰 번호, SNS 등 아무런 정보가 없어** 도무지 연락할 길이 없었다.

오명진 팀원은 2021년도 중앙대학교 소프트웨어학부 학술동아리 ChAOS의 회장을 맡았다. 회장
으로 활동할 당시, 기가 막힌 해커톤 대회 기획안이 떠올랐지만 회장단 친구들과 대회 진행에 관
련해 상의를 하면서 전문 디자이너의 부재에 관한 걱정이 들었다. 디자인 학과의 과동아리와 연
합해 해커톤을 진행해보고자 했으나 **연락을 취해 연합하기가 쉽지 않았고**, 해커톤 대회는 결국
흐지부지 되었다.

우리 팀원들은 이렇게 직접 느낀 문제점과 경험을 바탕으로 중앙대학교 내 **과동아리 간 연합 활동 및 교류 네트워크 구성**을 고안해보게 되었고, 이 어플리케이션을 기획하게 되었다.

II. 기존 서비스들과의 비교 / 분석

학생 커뮤니티, 동아리, 공모전, 대외활동 등 즐겁고 유익한 정보를 한 곳에 모아 제공하는 서비스인 대형 플랫폼 '캠퍼스픽'과 'CAUCLUB'을 해당 서비스들의 목적과 주된 기능인 연합(공모전, 스터디, 대외활동 모집 등), 랭킹 제도로 나누어 조사를 진행했다.



1) 공모전 스터디, 대외활동 모집

- 유사점

함께 활동할 사람을 구한다.

- 차별점

캠퍼스픽은 동아리와 무관하게 개개인이 모집을 진행하는 반면, CAUCLUB은 각 동아리와 동아리 간에서 모집이 이뤄지며, 이에 따라 활동 측면에서는 보다 체계적이고 조직적인 기획, 진행이 가능하며 동아리 운영의 측면에서도 해당 활동 참가 대상의 모집, 홍보 등이 용이하다.

2) 랭킹 제도

- 유사점

분류별로 동아리 랭킹 제도를 운영하고 있다.

- 차별점

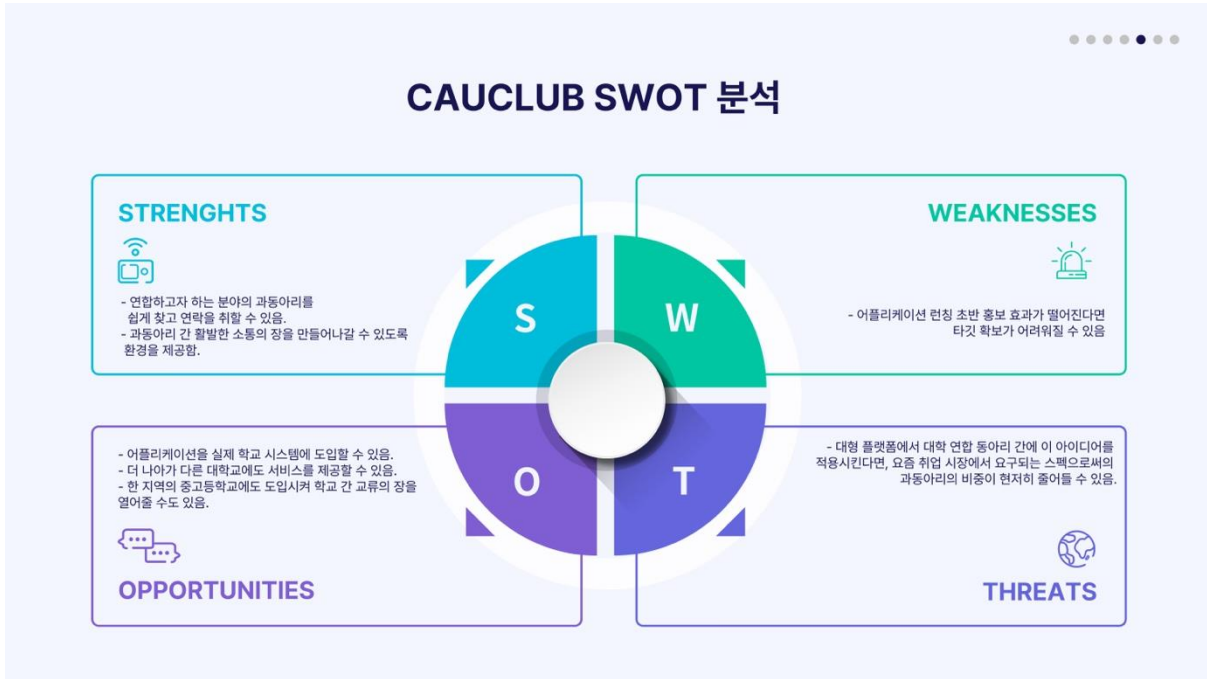
캠퍼스픽에서는 **좋아요 개수만을 기준으로** 랭킹을 매기고 있으며, 이러한 방식은 실제 해당 동아리의 활동과 무관하진 않지만 단순 인기, 인력, 규모만으로 얻을 수 있기에 활동의 질과 연관성이 떨어진다는 단점이 있다. 활동을 멈춘 동아리가 발생하면 상단 노출이 의미가 없어지며, 활동을 멈추더라도 좋아요를 받을 수 있기에 6개월마다 초기화가 이뤄지는 방식으로 되어있다. 앞서 말한 특성에 따라 동아리를 운영할 때에 활동보다는 동아리 자체의 규모를 키우고 이를 위한 홍보에만 치중할 수 있다는 문제점이 있다. **CAUCLUB에서는 좋아요 개수 보다는 실제로 이루어진 활동에 대해서 훨씬 더 큰 랭킹 점수를 획득**할 수 있고, 따라서 활발하게 다양한 활동을 하며 동아리 간의 연합, 소통을 목적으로 하는 동아리의 랭킹이 높아지게 된다. 이로 인해 **각 동아리는 단순 규모나 홍보가 아닌 실질적인 활동에 집중**하게 되는데, 동아리에서 활동에 집중해 점수가 높아지면 자연스럽게 랭킹의 상단에 노출되며 홍보의 효과를 얻고 규모가 커지며, 이러한 선순환이 계속될 수 있다. 우리의 서비스에는 초기화 시스템이 도입되어 있지 않기 때문에 활동을 멈춘 동아리가 발생하는 경우 해당 동아리의 점수는 정체되며, 다른 동아리의 점수는 계속해 올라가기에 초기화가 필수적이지 않다는 장점도 있다.

캠퍼스픽을 제외한 다른 기존 유사 서비스들인 링커리어, 캠퍼즈, 마이캠퍼스와의 비교, 분석은 아래의 차트를 통해 정리했다.

	LINKareer 링커리어	CAMPUZ 캠퍼즈	마이캠퍼스	CAUCLUB
동아리 규모	연합동아리			
	대학 과동아리			
랭킹 제도				
아카이브				
채팅				
모집 대상	개인			
	동아리			

III. 설계

1) SWOT 분석



전략적으로 프로젝트 계획을 세우고 시장 트렌드에 앞서고자 **SWOT 분석**을 활용하였다. SWOT란 **Strengths**(강점), **Weaknesses**(약점), **Opportunities**(기회), **Threats**(위협)을 의미하며, 이 4가지 SWOT 프레임워크를 함께 분석함으로써 현재 상황과 다음 단계로 나아가기 위해 어떻게 해야 하는지 거시적으로 파악할 수 있었다.

2) 기능 설계

• 회원가입

아이디, 이름, 이메일, 소속 대학의 기본 정보를 입력하고, 동아리 추천을 위한 키워드를 선택해 가입이 가능하다.

• 아이디, 비밀번호 찾기

사용자가 아이디 / 비밀번호를 분실했을 때, 가입한 이메일을 통해 랜덤으로 생성된 6자리 인증번호를 받고, 해당 번호를 통해 인증하게 된다. 인증 성공 시, 아이디 찾기의 경우 아이디를 화면을 통해 보여주게 되고, 비밀번호 찾기의 경우 재설정 창으로 이동하게 된다.

• 동아리 생성 및 동아리 가입/탈퇴

사용자는 자신의 학과 내의 동아리를 생성할 수 있고, 같은 학과의 동아리에 가입 및 가입 취소를 할 수 있다.

- **사용자가 설정한 키워드를 통한 동아리 추천**

사용자의 성향에 따라 선택한 키워드들과 가장 부합하는 다른 학과의 동아리들을 추천 받을 수 있다.

- **동아리 게시판**

동아리 회장은 동아리 게시판에 글을 써서 연합 활동 동아리 모집을 홍보할 수 있다.

- **동아리 아카이브**

동아리 회장은 해당 동아리의 활동 및 타 학과 동아리와의 연합 활동을 기록할 수 있는 아카이브 기록 페이지를 사용할 수 있다.

- **채팅**

사용자는 소통하고 싶은 동아리의 프로필을 통해 해당 동아리장과 채팅을 시작할 수 있다.

- **동아리 랭킹**

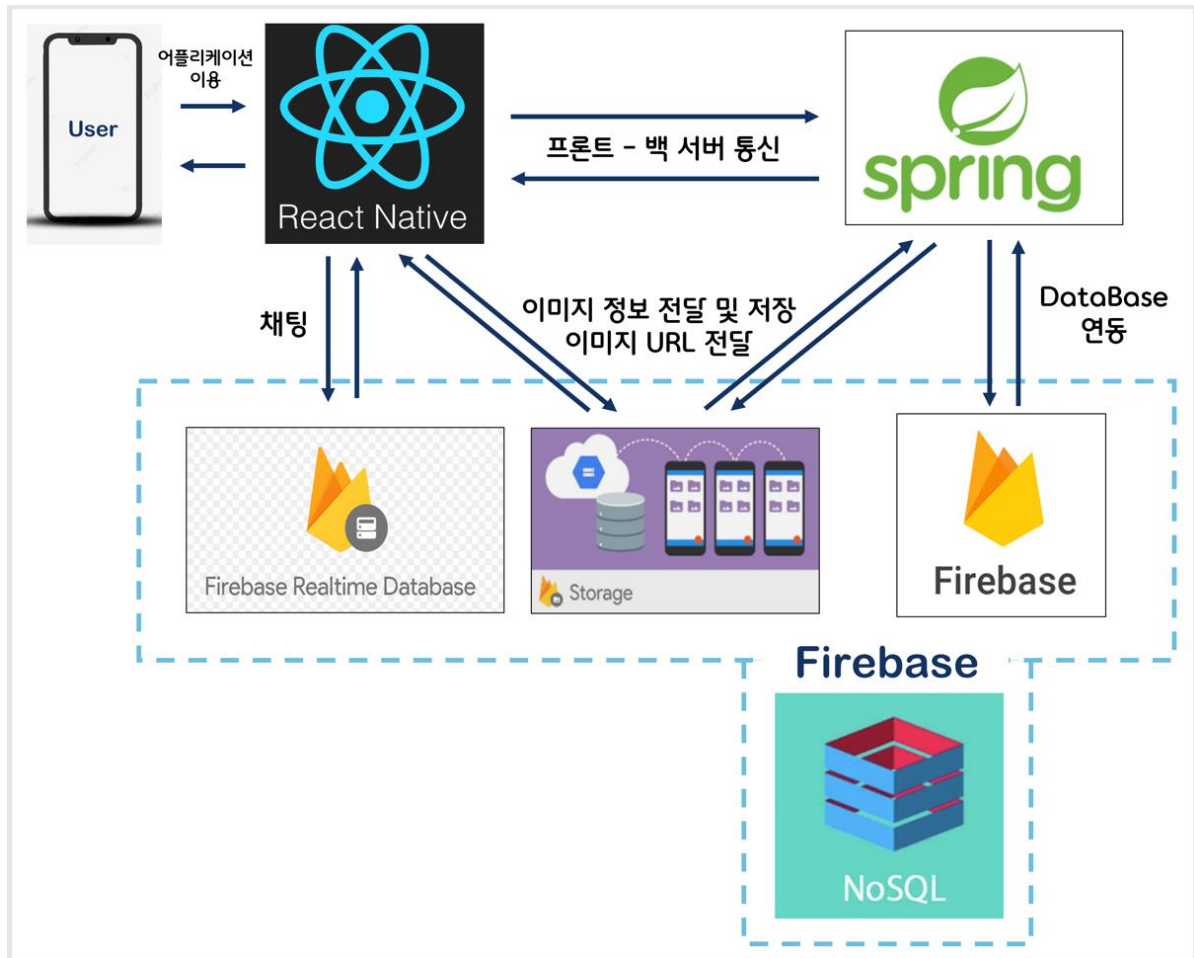
활동 점수를 기준으로 전체 동아리 / 학술 동아리 / 예체능 동아리 / 기타 동아리의 전체 랭킹을 볼 수 있다.

- **동아리 회장 위임**

각 동아리 회장은 그 다음 동아리 회장의 아이디를 입력함으로써 회장 권한을 위임할 수 있다.

3) 기술 설계

전반적인 기술 흐름은 다음의 도식을 통해 정리하였다.



세부적인 기술의 설계사항은 다음과 같다.

• Front-End

Front-End는 오픈소스 모바일 애플리케이션 프레임워크인 **React-Navite**를 선택했다.

동일한 코드로 ios와 android 환경 모두에서 네이티브 앱으로 빌드가 가능하며, react에서 파생되어 컴포넌트 기반으로 높은 재사용성을 지니며 오픈소스이기에 많은 모듈과 자료를 찾아볼 수 있어 선택하게 되었다.

사용된 여러 모듈 중 핵심적인 모듈로는,

- ios의 노치 디자인으로 인한 안드로이드 화면과의 불일치를 해소하기 위한 **SafeAreaView**,
- Back-End의 REST API 와의 네트워크 통신을 위한 **axios**,
- 채팅, 이미지 업로드를 위한 firebase 연동을 위한 **firebase** 관련 모듈들,
- pbkdf2 방식의 비밀번호 암호화를 위한 **crypto**
- 모바일 화면에서의 자연스러운 화면전환과 상단, 하단바를 위한 **navigation** 이 있다.

유사한 형태로 세부적인 정보만 바뀌며 반복적으로 렌더링 되는 게시글, 아카이브 글, 프로필 등을 컴포넌트로 분리해 재사용성과 유지보수를 용이하게 하는 방향으로, **선언적 프로그래밍**을 사용했다.

기존 웹/리액트 프로젝트에서 흔히 사용되는 javascript의 경우 동적 타이핑으로 인해 개발 시, 사용시 타입 안정성이 보장되지 않기에 type strict하게 **typescript**를 사용하기로 하였다. 여러 부분에서 사용되는 동아리, 사용자 등의 추상적인 객체를 type을 통해 지정하여 개발하며 더욱 안정적인 개발이 가능했다.

Firebase를 데이터베이스로 사용하며 firebase의 자체적인 한계로 데이터를 받아오는 시간을 줄이기 어려워 데이터를 받아오거나 시간이 필요한 처리에 있어서 async, await, promise를 이용한 **비동기적** 프로그래밍을 적극 사용하였다.

• Back-End REST API

Back-End는 java 웹 프레임워크인 spring을 spring boot 라이브러리를 통해 설정하는 방식으로 구성하였으며, NoSQL 기반의 firebase database를 사용하였다. Firebase database의 경우 NoSQL 기반이기에 구조를 변경하기 쉬우며, 기능을 점진적으로 추가하기 좋으며, 많은 기능이 제공되어 선택하게 되었다

• Database

User	Archive	Club	Post	CollectionPrimaryKey
+ id + password + email + department + joinedClub + managingClub + keyword + name + salt	+ archiveId + clubId + title + contents + createdAt + modifiedDate + reportCount + reportMemberList + like + likeMember + isMutual	+ clubId + department + keyword + leaderId + score + type + reportCount + introduction + members + picture	+ postId + clubId + title + contents + createdAt + modifiedDate + reportCount + reportMemberList	↯ club + archive + post

채팅의 경우 firebase의 Realtime Database에 JSON snapshot으로 저장되고 읽히며, 형태는 다음과 같다.

```
{
  "chatService": {
    "rooms": { #채팅방
      "-N3s7Ckm3dGb96FFVpFw": { #채팅방 id
        "chats": {
          "-N3s7Vo_CLogmRpFcR3C": {
            "sentBy": "user1", #보낸 사람
            "text": "채팅1" #채팅 내용
          },
          "-N3s7ZSo1HsoKA1L5HAU": {
            "sentBy": "user2",
            "text": "채팅2"
          },
          "-N3s7ab8_J-4JSSF5dkf": {
            "sentBy": "user1",
            "text": "채팅3"
          },
          "-N3s7dQZWyuF5F_oBJdI": {
            "sentBy": "user2",
            "text": "채팅4"
          }
        }
      }
    },
    "userRooms": {
      "user1": { #로그인한 유저
        "user2": { #대화 상대
          "opName": "user2Name", #대화 상대의 이름 (화면 표시 이름)
          "picture": #채팅방 사진
            "https://firebasestorage.googleapis.com/443/v0/b/caclub1.appspot.com/o/IMG_0002.JPG?alt=media",
          "recent": "채팅4", #최근 메시지
          "roomId": "-N3s7Ckm3dGb96FFVpFw" #채팅방 id
        }
      },
      "user2": {
        "user1": {
          "opName": "user1Name",
          "picture":
            "https://firebasestorage.googleapis.com/443/v0/b/caclub1.appspot.com/o/IMG_0002.JPG?alt=media",
          "recent": "채팅4",
          "roomId": "-N3s7Ckm3dGb96FFVpFw"
        }
      }
    }
  }
}
```

IV. 과제 수행 과정 및 구성원 간 역할분담

1) 개인별 과제 수행 결과

- **이기태**

- Firebase Firestore Database 구조 정립
- spring boot를 이용한 Backend REST API 구현
- 추천 시스템 설계 및 구현
- 데모를 위한 예시 Database 구축

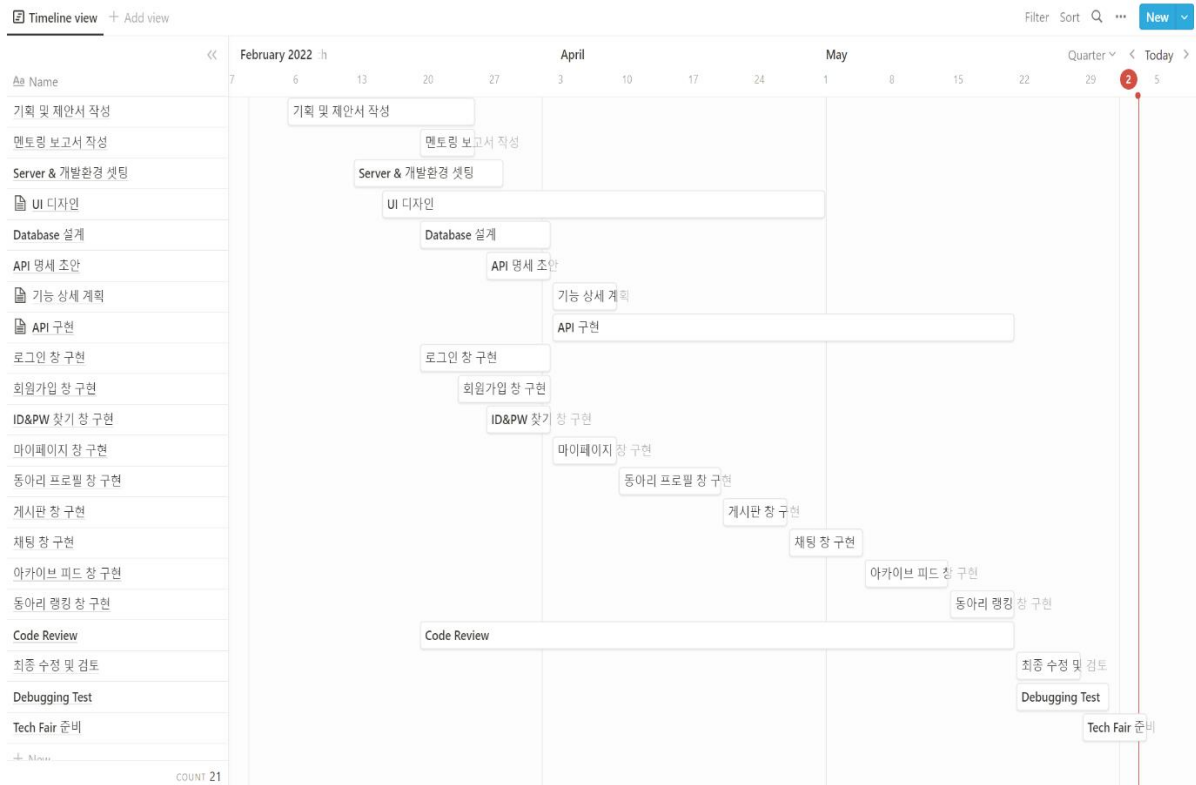
- **오명진**

- React Native의 jsx 문법으로 작성된 화면의 view가 동작하도록 창의 모든 로직 구현
- 구현된 REST API와 소통하여 application의 기능 구현
- Firebase Storage와 spring boot를 이용한 이미지 업로드 API 구현
- Firebase Realtime Database를 이용한 채팅 서버 구현
- 모바일 화면 흐름을 위한 navigation 구현

- **이예빈**

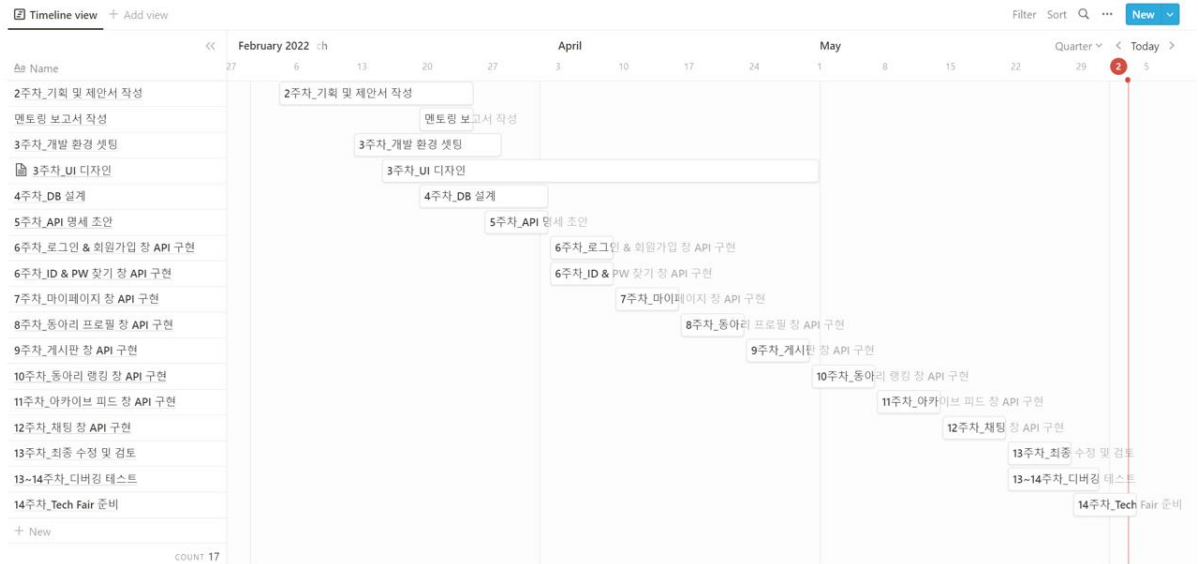
- Figma를 이용하여 Frontend 화면 view UI/UX 디자인
- application에 필요한 모든 view를 React Native의 jsx 문법으로 설계 및 구현
- Stylesheet component 설계 및 구현
- 데모를 위한 예시 DataBase 구축
- 기획 설계 Diagram 제작 및 문서 정리

2) 팀 스케줄

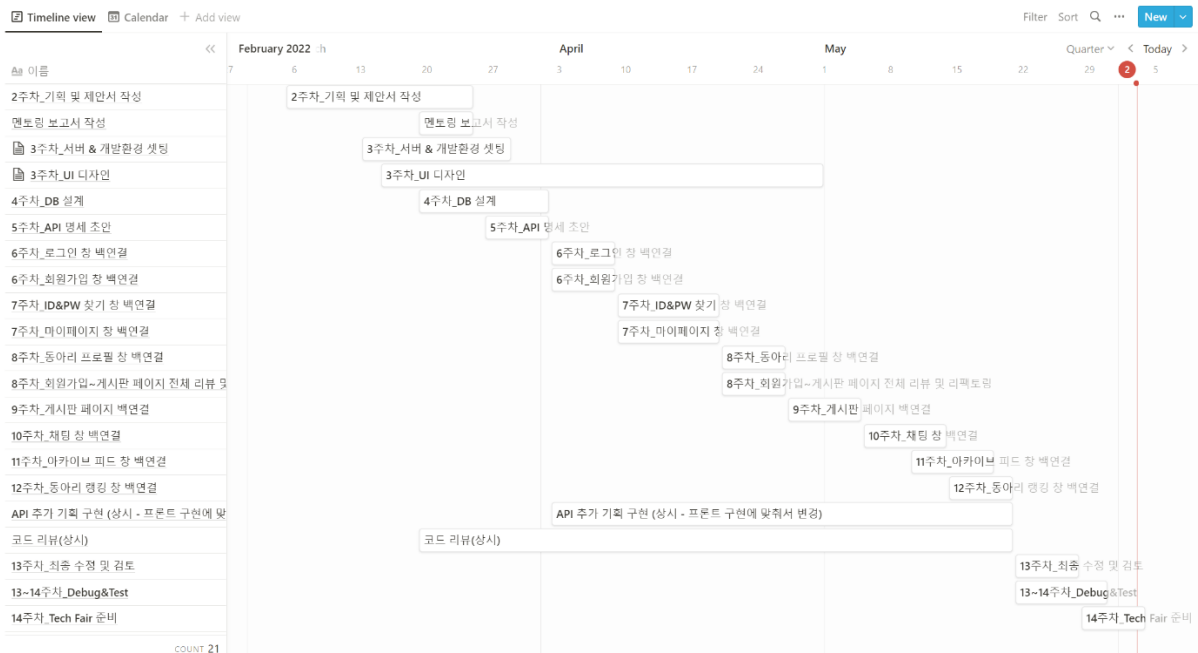


3) 개인별 스케줄

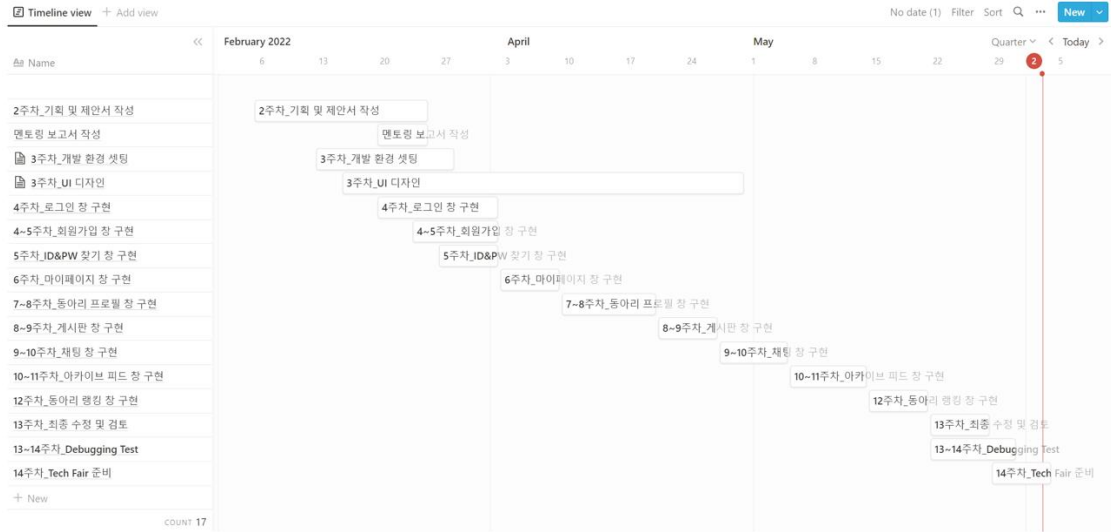
● 이기태



● 오명진

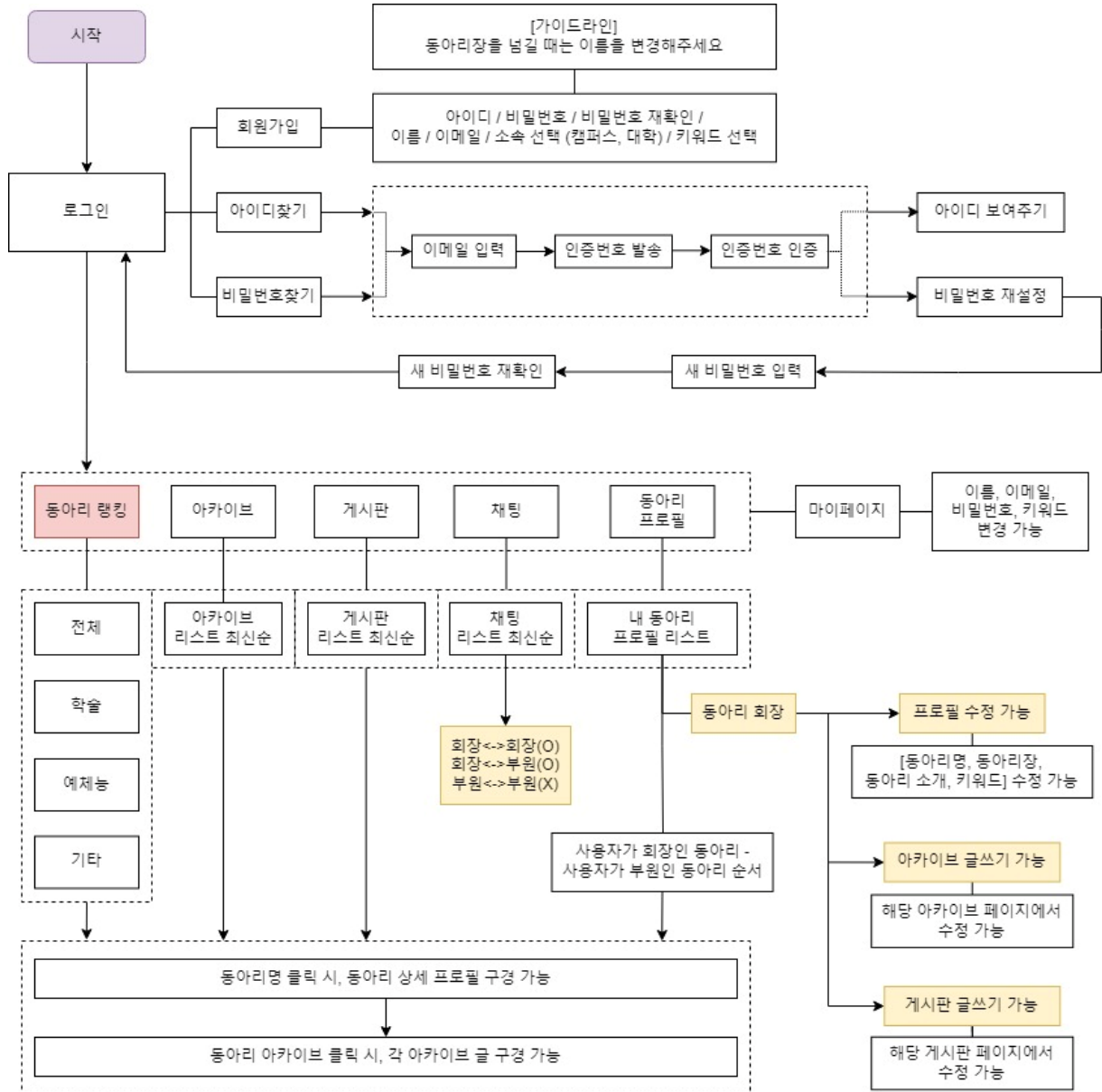


• 이예빈

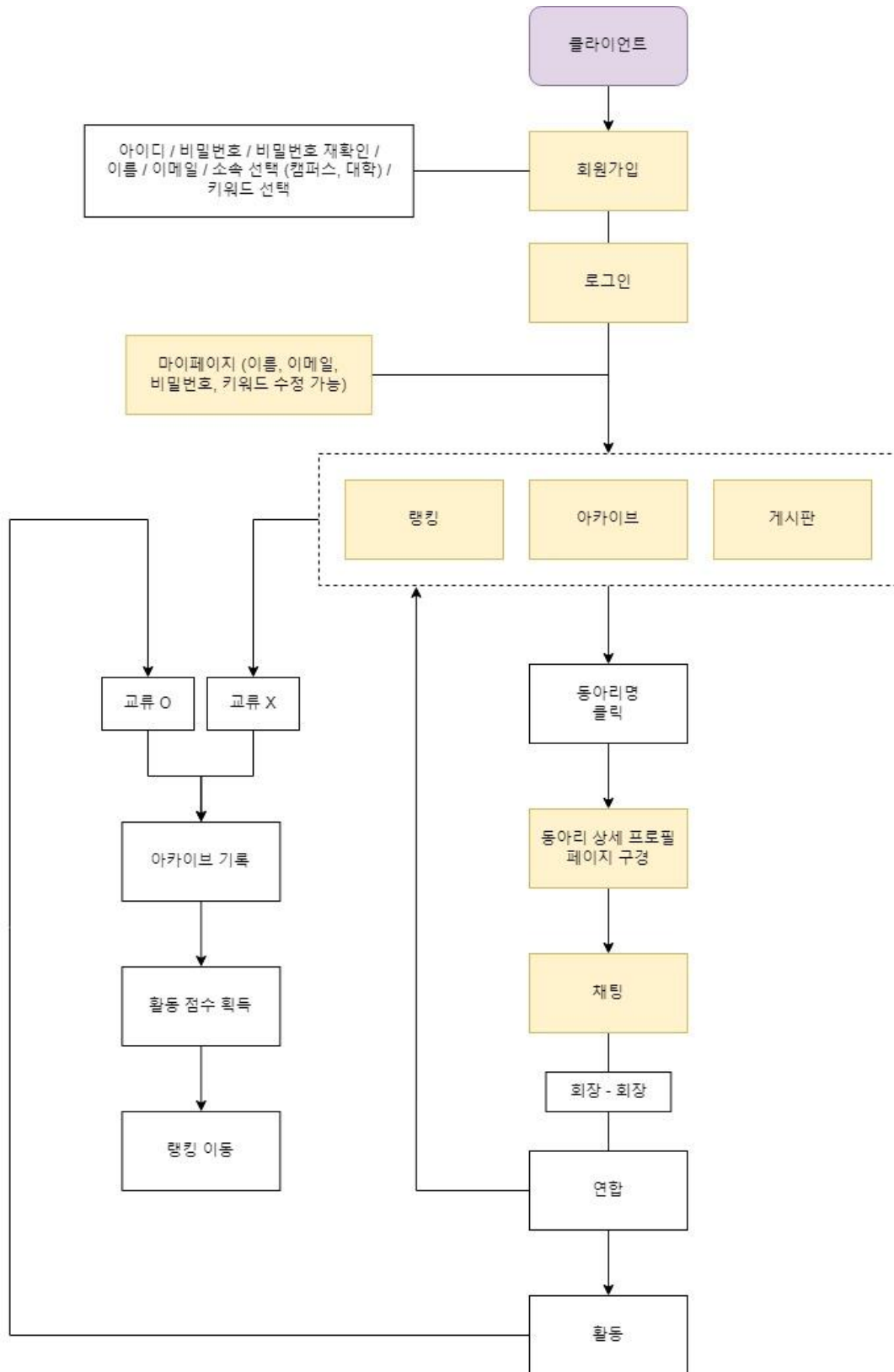


V. 기능 상세

1) 전체 기능에 대한 흐름도



2) 사용자 관점에서의 어플리케이션 화면



위 흐름도는 기존 제안서에 담았던 모든 내용을 구현한 결과를 나타낸 것이다. 처음 기획했던 것 외에도, 우리 팀은 '**신고 제도**' 기능을 도입하게 되었다. 이 기능을 통해 사용자는 게시판 혹은 아카이브 글이 부적절하다고 생각되면 신고하기 버튼을 눌러 해당 글을 신고할 수 있다. 사용자는 하나의 게시판 혹은 아카이브 글에 한번 씩만 신고 가능하며 중복 신고는 불가능하다. 신고 버튼을 누르면 해당 게시판 혹은 아카이브 글은 신고 누적 횟수가 1회 증가하고 동시에 글을 쓴 동아리의 신고 누적 횟수 또한 1회 증가한다. 글의 신고 누적 횟수가 3회가 되면 해당 글은 삭제 조치 되고 해당 아카이브 글로부터 받은 동아리 랭킹 점수는 다시 차감된다. 또한 동아리 신고 누적 횟수가 5회가 되면 동아리 랭킹 점수가 30점 차감되고 신고 횟수는 다시 0회로 초기화된다.

VI. 기술 특징

1) 채팅 서비스

실시간 상호작용을 위해 Firebase의 **Realtime Database**를 선택했다. JSON형식의 snapshot을 통해 미리 정해진 schema 없이 빠르게 실시간으로 데이터를 송수신 할 수 있다. 별도의 웹 프레임워크를 거치지 않고 프론트의 react-native-firebase를 통해 사용한다.

2) 비밀번호 암호화

로그인과 회원가입 과정에서 **비밀번호를 안전하게 저장**하기 위해 평문으로 데이터베이스에 저장하는 것이 아닌 **암호화된 값을 저장**하였다. 프론트에서 암호화 한 후 back으로 전송해 중간의 패킷 탈취로 인한 비밀번호 유출을 예방하였다. 암호화는 **PBKDF2 방식**으로 **SHA512 알고리즘**을 사용하였다. 매번 같은 해시 값이 도출되어 결과만 보고 원문을 유추할 수 있는 레인보우 테이블을 쉽게 만들지 못하도록 **salt 값**을 통해 무작위성을 부여하였고, 여러 번의 iteration으로 보안을 높였다.

3) 추천 시스템

사용자가 회원가입 시에 선택한 키워드들을 각 동아리가 지정해 놓은 키워드들과 비교한다. 겹치는 키워드가 많은 순서대로 동아리를 정렬한다. 정렬된 동아리 중 상위 5개를 사용자에게 추천해주며 소통의 가능성을 높인다

VII. 활용방안 및 기대효과

1) 과동아리 간 소통의 어려움 해결

본래 과동아리의 경우 대부분 해당 학과 내에서만 활동을 하기에 연합동아리나 중앙동아리에 비해 다른 동아리와의 소통이 쉽지 않다. CAUCLUB을 통해 소통을 원활하게 함으로써 과동아리 간의 네트워크를 형성하고, **보다 활발한 소통의 장**을 만들 수 있다.

2) 과동아리의 전문성 향상

원활한 이해를 위해 소프트웨어학부 학술동아리 ChAOS의 예시를 들어 설명을 해보고자 한다. 프로젝트를 진행할 때, 디자인적 요소에 너무 집중하다 보면 개발에 전념하지 못하는 경우가 있을 수 있다. 디자인 학과와 연합하여 각자의 역할에 충실한다면 더 나은 결과를 이끌어내며, 각 동아리 분야의 전문성을 극대화 시켜 작품을 만들어낼 수 있을 것이다.

3) 지역 내 중고등학교 동아리 연합 형성 가능성

현재 지역 내의 중고등학교 간의 동아리 연합이 많이 부족한 상황이다. 다양성을 추구하기 위해 교육청에서도 학교 간 공문 발송을 통해 선생님들의 주도 아래 동아리 연합을 추진하고 있다. 요즘은 학생들이 SNS에 많이 참여하는 시대이기에 CAUCLUB을 이용해 학생들이 주도적으로 동아리 연합 활동을 기획해본다면 리더십 형성, 사회성 발달 등에 도움이 될 것이다. CAUCLUB 하나의 아이디어로 여러 다양한 긍정적인 결과를 기대해볼 수 있다.

VIII. 향후 발전 방향성

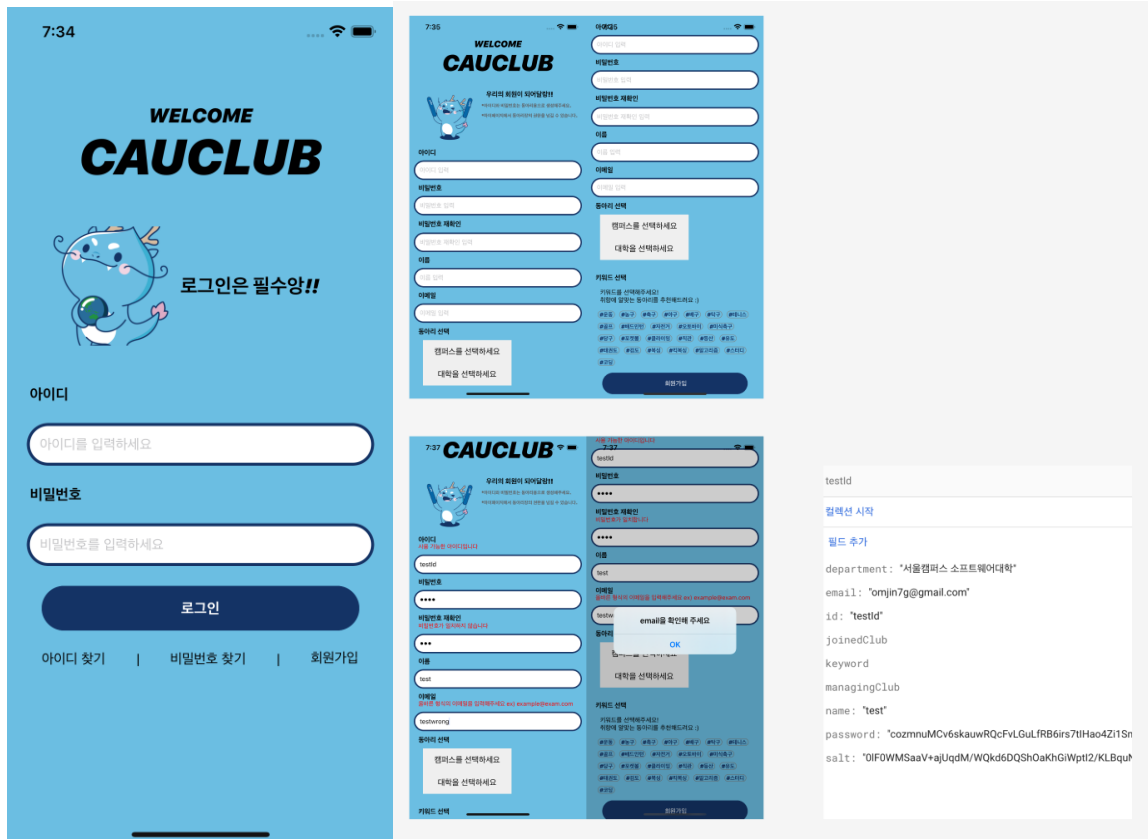
우리 팀은 어플리케이션 기능을 완성한 것에 그치지 않고 추가 구현했으며 더 나아가 **향후 발전 방향성**에 대해서도 고려해보았다. 그 아이디어들은 다음과 같다.

- 이전에 [세부내용]에서 '신고 제도' 라는 기능을 추가 구현했다고 소개하였다. 부적절한 글을 올리거나 원활한 소통을 방해하는 사용자들을 방지하고자 하는 목적으로, 게시판 및 아카이브 글쓰기 기능은 동아리 회장만 이용 가능하도록 하였다. 다만, 채팅 기능은 회장과 회장, 회장과 부원 간 소통은 가능하나 부원 간의 소통은 불가능하도록 설계 및 구현을 완료한 상태다. 하지만 뒤늦게 제안서 외의 추가 기능 구현으로 신고 기능을 도입했고 이 기능을 도입함으로써 사용자들 간 소통을 확대해도 좋을 것 같다는 생각을 했다. 즉, 동아리 부원들끼리도 서로 채팅 할 수 있게 기능을 확장할 예정이다.
- 앞서 생각한 아이디어의 연장선이다. 채팅 기능을 조금 더 확장해 같은 동아리 내의 여러 부원들끼리 서로 소통이 가능하도록 **'동아리 단체 채팅'** 기능을 추가할 예정이다. 현재 동아리 단체 채팅방은 주로 카카오톡을 통하여 형성된다. 그러나 평소 동아리 부원들의 연락처를 모두 받아 카카오톡 단체 채팅방을 형성하고 관리하는데 어려움을 느끼는 경우가 많았고, 이 아이디어를 도입하여 이 문제점을 해소함과 동시에 사용자를 대다수 확보할 수 있다.

- 현재는 게시판과 아카이브 글에 댓글 및 대댓글 기능이 존재하지 않는다. 그러나 앞에서 설명했듯이, 신고 기능 제도가 도입이 되었고 CAUCLUB은 실명제라는 점에서, 댓글 및 대댓글 기능을 추가해도 큰 문제없이 사용자들끼리 즐거운 소통이 가능하리라 생각된다.
- [세부 내용]에서 우리 CAUCLUB 프로젝트의 SWOT 분석에 관해 설명하였다. 이 중 **Weaknesses**(약점)으로 어플리케이션 런칭 초반에 홍보 효과가 떨어진다면, **타겟 확보가 어려워질 수 있다는 문제점을 스스로 지적**하였다. 이를 해결할 방안을 고민해보았고 다음과 같은 아이디어를 생각해볼 수 있었다. 중앙대학교 서울캠퍼스 정문 흑석역 근처 상인들과 연합을 맺어 해당 가게의 상품권을 할인 판매하는 기능 등을 도입함으로써 커뮤니티 및 지역 경기 활성화에도 도움이 될 수 있고, 이런 메리트를 이용하고자 CAUCLUB 어플리케이션을 이용하려는 동아리가 늘어나게 될 것이라 판단하였다.

IX. 결과물

1) 앱 시작 화면, 회원가입 및 가입 결과



회원이 가입 버튼을 통해 회원가입 창으로 이동이 가능하며, 회원가입이 완료되면 로그인 창으로 돌아가게 된다. 가입한 결과가 Firebase의 Firestore database에 잘 저장된 것을 확인할 수 있다.

2) 아이디 찾기 및 실행 결과



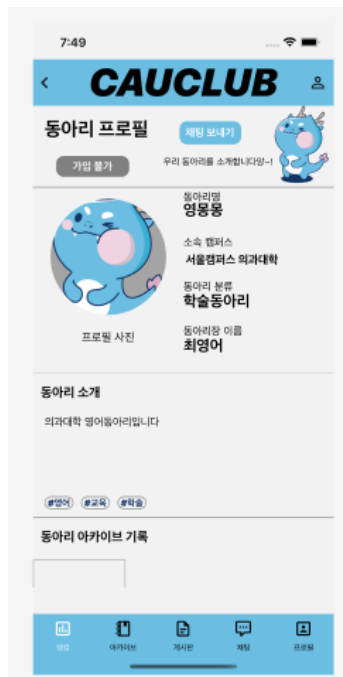
3) 비밀번호 찾기



4) 분류별 동아리 랭킹



5) 타과 동아리 프로필



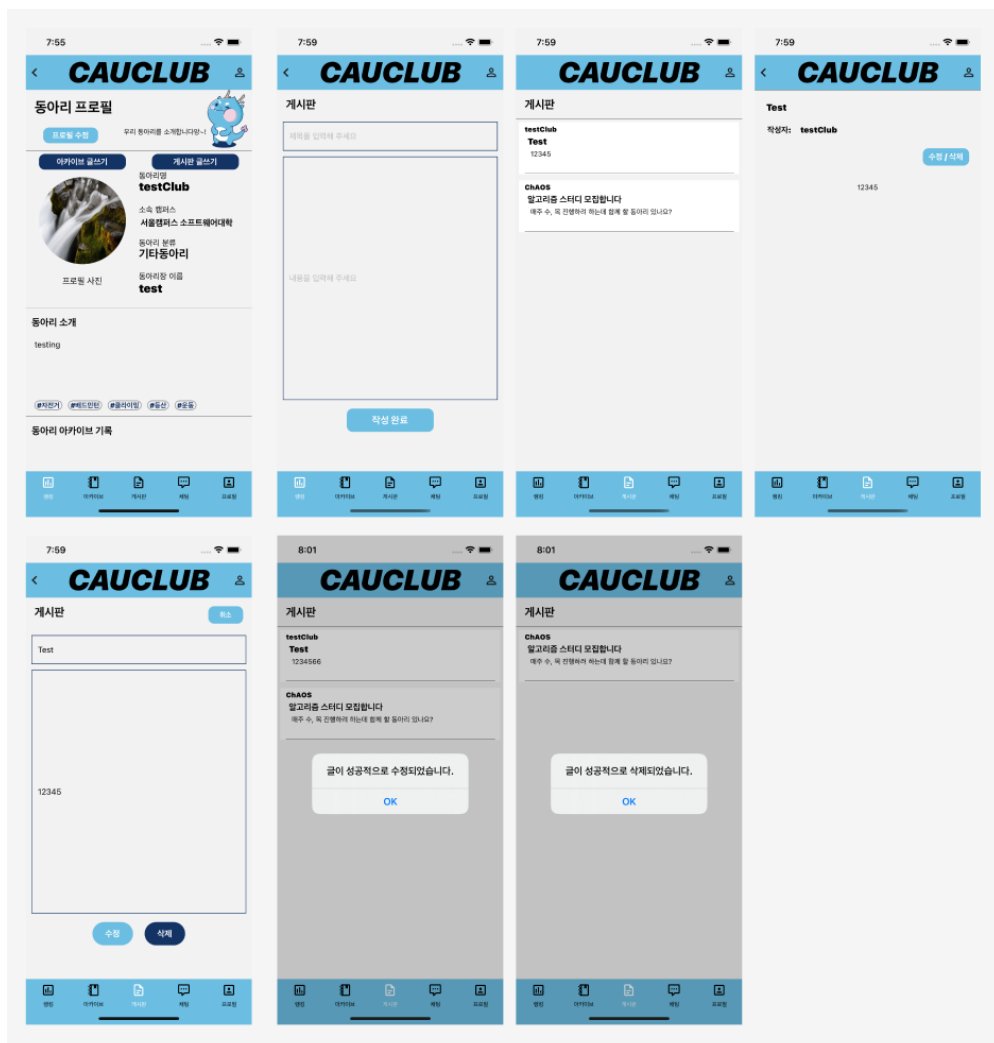
6) 자과 동아리 프로필 – 가입 / 탈퇴



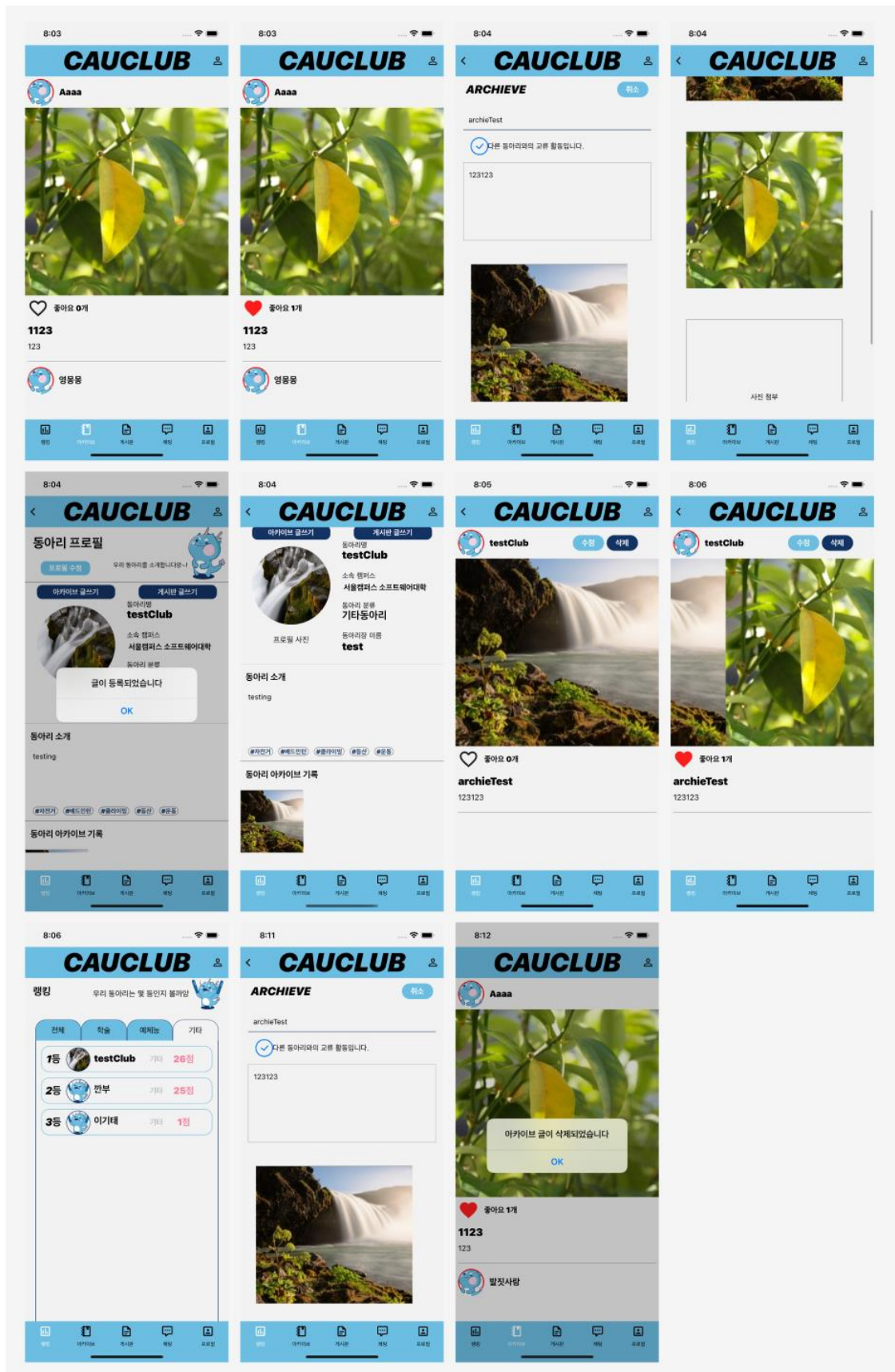
7) 동아리 생성



8) 게시판 글 작성, 수정, 삭제



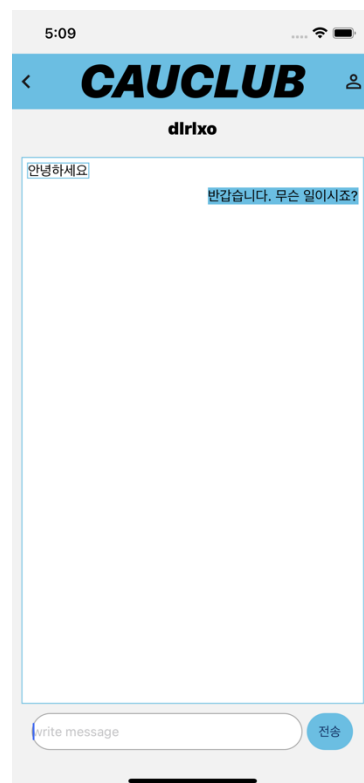
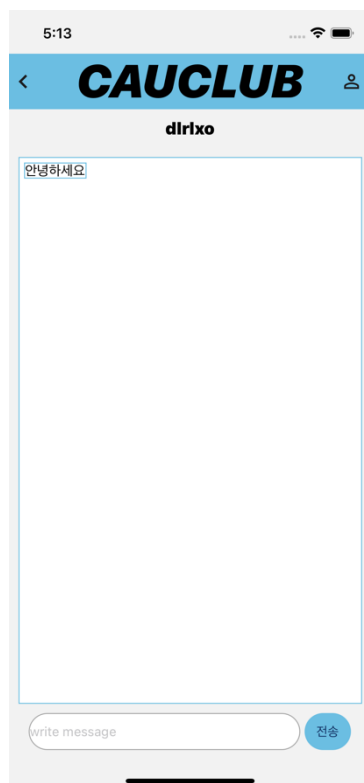
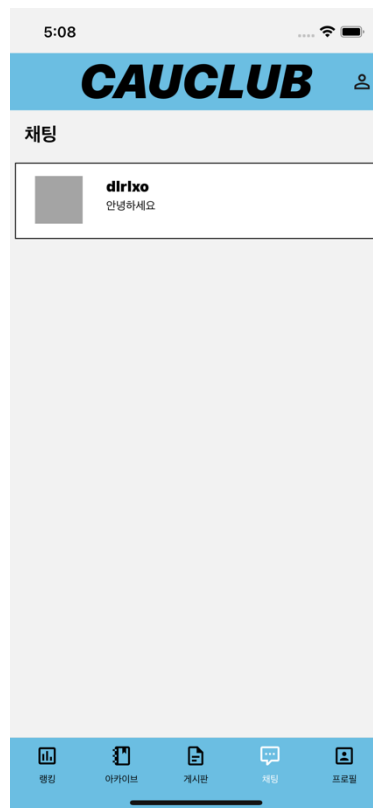
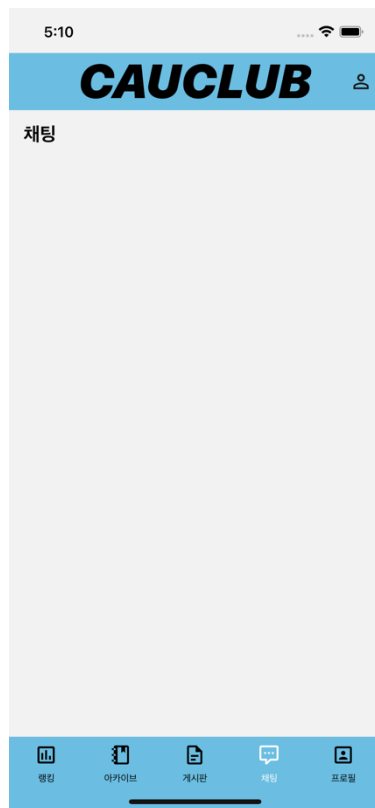
9) 아카이브 글 작성, 좋아요, 수정, 삭제



10) 동아리 추천 -> 마이페이지에서 키워드 수정 -> 재추천



11) 채팅 : testId 유저와 ChAOS 동아리 회장 간의 채팅



12) 로그아웃, 회원 탈퇴



13) 아카이브 및 게시물 신고

