



국민대학교
전자정보통신대학
컴퓨터공학부

관리번호 : 20xx-XX

캡스톤 디자인 I

종합설계 프로젝트

프로젝트 명	디클
팀 명	7조 디클
문서 제목	최종 보고서

날짜	2024 년 06 월14일
----	----------------

팀원	권 지아(조장)
	윤 흥현
	윤 응배
	김 동운
지도교수	이 창우 교수

캡스톤디자인
|
(2024-
07)

디
클

국민대
교
학
소프트
웨어
학
부

제 출 문

본 보고서를 캡스톤 디자인 I 교과목의 종합설계 프로젝트
최종 보고서로 제출합니다.

2024. 06. 14

프로젝트 명 : 디클

팀 구성원 : 권 지아 (팀장)

윤 흥현

윤 웅배

김 동윤

지도교수 : 이창우 교수

목 차

1. 요약
2. 계획서
 - 2.1 계획서
 - 2.2 계획서 발표자료
3. 중간보고서
 - 3.1 1차 중간보고서
 - 3.2 1차 중간보고서 발표자료
 - 3.3 2차 중간보고서
 - 3.4 2차 중간보고서 발표자료
4. 결과보고서
 - 4.1 결과보고서
 - 4.2 결과보고서 발표자료
5. 회의록
6. 프로젝트 수행 자체 평가

제 1 장

요약

디클은 전국 대학생들이 학과를 중심으로 모여 소통할 수 있는 학과별 커뮤니티 어플이다. 기존 에브리타임과 같은 커뮤니티는 소속 학교를 중심으로 모이므로, 같은 전공 분야에 있는 다양한 학교의 학생들과는 소통이 어렵다는 문제가 있다. 또한 에브리타임 내의 학과별 게시판도 활성화 되어있는 일부 학과들 외에는 사실상 사용하는 사람이 적다. 따라서 전국의 모든 대학교 재학생들이 학과별로 모여 정보를 공유하고 자유롭게 소통할 수 있는 커뮤니티를 개발하고자 한다.

Decl is a departmental community app that allows college students across the country to gather and communicate around their majors. Existing communities like Everytime are centered around your school, so it's difficult to communicate with students from various schools in the same major. Also, the departmental bulletin boards within Everytime are rarely used except for some departments that are active. Therefore, we want to develop a community where students from all universities across the country can gather by department and share information and communicate freely.

제 2 장

계획서

2.1 계획서

2.2 계획서 발표자료

제 3 장

중간보고서

3.1 1차 중간보고서

3.2 1차 중간보고서 발표자료

3.3 2차 중간보고서

3.4 2차 중간보고서 발표자료

3.1 1차 중간보고서

목 차

1	프로젝트 목표	4
2	수행 내용 및 중간결과	5
2.1	계획서 상의 연구내용	5
2.2	수행내용	5
3	수정된 연구내용 및 추진 방향	6
3.1	수정사항	6
4	향후 추진계획	7
4.1	향후 계획의 세부 내용	7
5	고충 및 건의사항	8

1. 프로젝트 목표

디클(Decl)은 전국의 대학생들이 학과를 중심으로 모여 소통할 수 있는 학과별 커뮤니티 어플입니다. 기존의 에브리타임과 같은 커뮤니티는 소속 학교를 중심으로 모이다보니, 같은 전공 분야에 있는 다양한 학교의 학생들과는 소통이 어렵다는 문제점이 있고, 또 에브리타임 내의 학과별 게시판도 활성화 되어있는 일부 학과들 외에는 사실상 사용 유저가 없는 상황입니다. 따라서 저희는 전국의 모든 대학교 재학생들이 학과별로 모여 정보를 공유하고 자유롭게 소통할 수 있는 커뮤니티를 개발하고자 합니다.

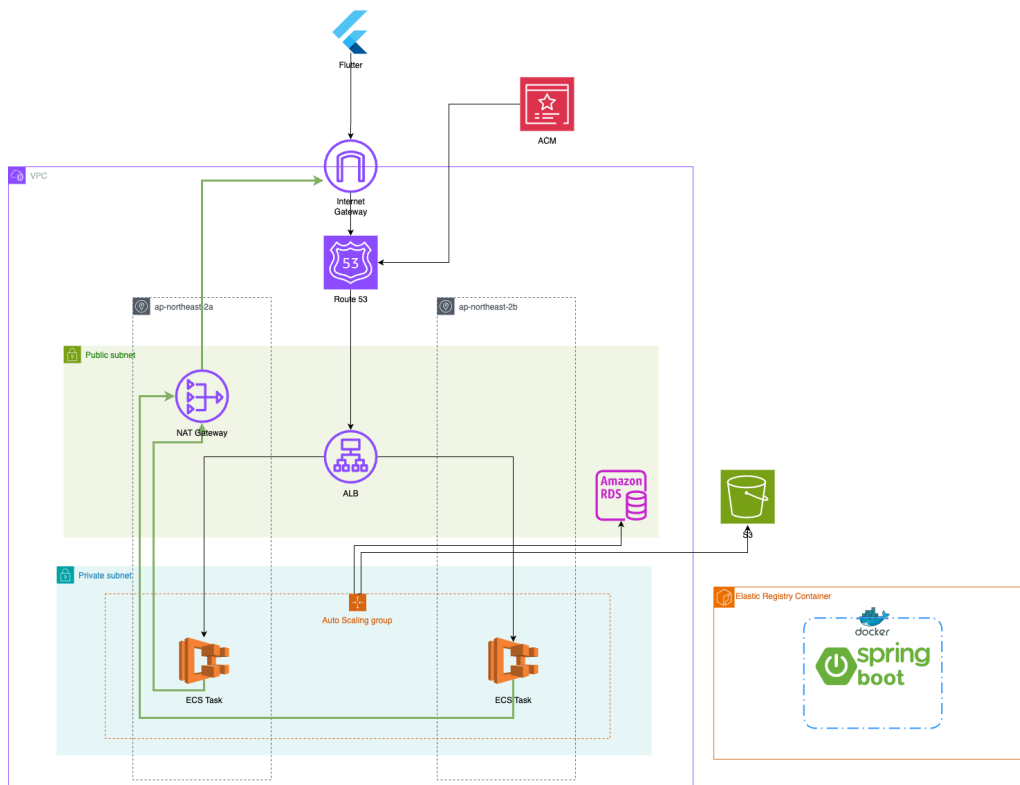
[프로젝트 사용 기술]

Frontend: Flutter 3.13.0, Dart 3.1.0

Backend : Spring Boot 3.2.3, Kotlin 1.9.0, Kotlin-jdsl 3.3.1

Infrastructure : Terraform, AWS ECS + Fargate

CI/CD : github action



2. 수행 내용 및 중간결과

2.1 계획서 상의 연구내용

학과별 커뮤니티 어플 '디클'의 주요 기능 4가지는 다음과 같습니다.

(1) 학교 이메일 인증을 통한 회원가입 및 로그인

전국 401개 대학교의 이메일 SUFFIX를 데이터베이스에 저장해 회원가입시 검증 및 이메일 인증을 진행합니다.

(2) 전공 최대 2개 선택 가능

회원가입시 최대 2개의 학과를 선택하여, 메인 탭에서 전환하여 본전공과 복수전공의 정보를 모두 챙길 수 있습니다. (주전공 및 복수전공 변경은 90일마다 가능)

(3) 전공 관심사에 따른 카테고리 분류

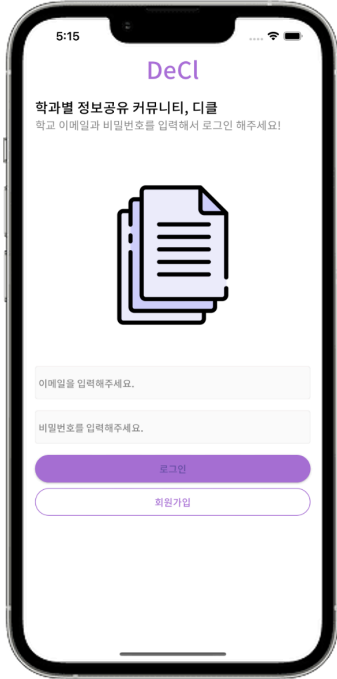
같은 전공 분야의 학우들과 이야기를 나누고 싶은 관심사에 따라 카테고리를 분류하고 필터링 기능을 제공합니다. (자유/대학원/취준/스터디/질문/홍보)

(4) 게시글 CRUD, 마이페이지 기능 등

커뮤니티 이용을 위한 게시글 및 댓글의 CRUD(Create, Read, Update, Delete), 마이페이지에서 확인이 가능한 작성글 조회, 댓글 단 글 조회, 스크랩 글 조회, 회원정보 변경 등의 기능을 제공합니다.

2.2 수행내용

진행 상황 : 30 개중 16 개 수행 완료 - 53% 수행 완료



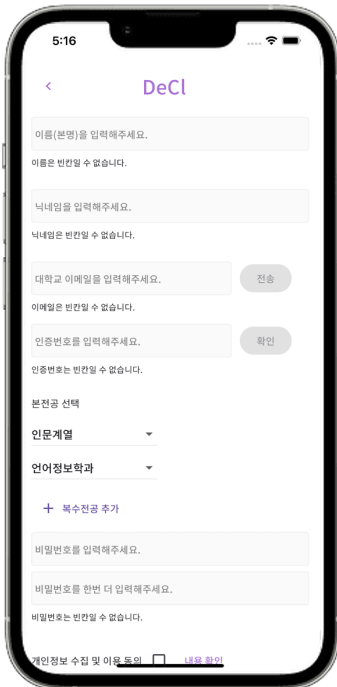
<로그인 화면>

[수행 완료]

- 로그인 화면 구성
- 로그인 기능 구현
- 회원 인증 상태 관리 기능 구현

[수행 미완료]

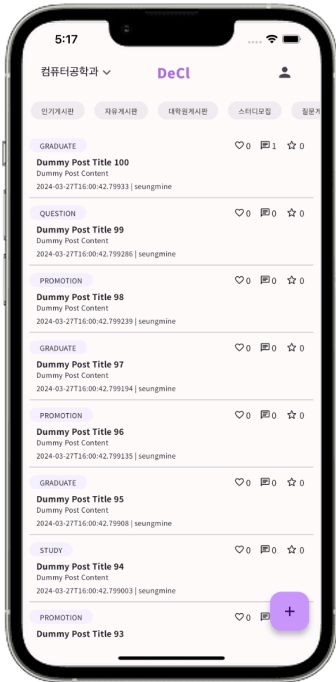
- 아이디/비밀번호 찾기 기능



<회원가입 화면>

[수행 완료]

- 회원가입 화면 구성
- 이메일 인증 기능 구현
- 회원가입 기능 구현



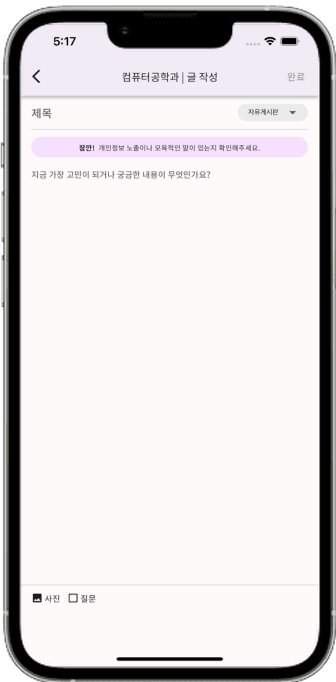
<글 리스트 화면/메인화면>

[수행 완료]

- 글 리스트 화면 구성
- 학과별 글 리스트 불러오기 기능 구현
- 리스트 pagination
- 카테고리 별 글 리스트 분류 기능 구현

[수행 미완료]

- 게시글 검색 기능



<글 작성 화면>

[수행 완료]

- 글 작성 화면 구성
- 갤러리에서 사진 불러오기 기능 구현
- 글 작성 기능 구현

[수행 미완료]

- 잘못 선택한 사진 선택하여 삭제하기 기능



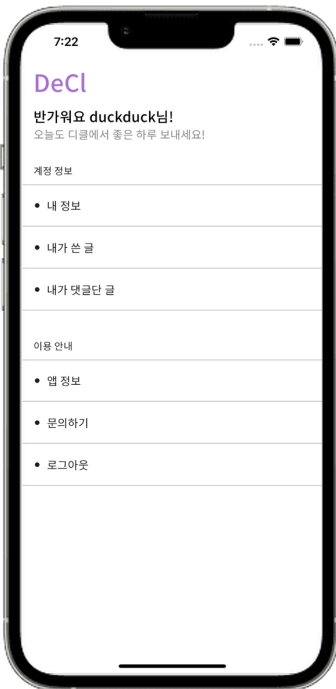
<글 상세 화면>

[수행 완료]

- 글 상세 화면 구성
- 글 상세정보를 불러와 적용
- 댓글 및 대댓글 달기 기능 구현

[수행 미완료]

- 좋아요/스크랩/알림 기능
- 게시글/댓글/대댓글 수정 및 삭제
- 댓글 페이지네이션



<회원정보 화면>

[수행 완료]

- 회원 정보 화면 구성

[수행 미완료]

- 내가쓴 글/댓글단 글/스크랩한 글 조회 기능
- 회원정보 수정 기능 (전공 변경, 비밀번호 변경, 회원 탈퇴)

3 수정된 연구내용 및 추진 방향

3.1 수정사항

1. 게시물 검색 기능 개발 및 조회 **api** 성능 개선: 사용자가 원하는 게시물을 검색할 수 있는 기능을 추가 개발할 예정입니다. 또한 조회 **api**를 호출하는 과정에서 발생하는 **select**문 등의 리소스를 점검하여 서버 성능을 최대한 개선할 예정입니다.
2. 사용자 활동 내역이 있는 게시물 조회 기능 추가 개발: 사용자가 작성한 게시물을 조회할 수 있는 기능을 추가 개발할 예정입니다. 뿐만 아니라, 사용자가 댓글 & 대댓글을 작성 및 스크랩한 게시물 또한 조회할 수 있는 기능을 제공하여, 사용자가 자신의 활동을 쉽게 추적하고 관리할 수 있도록 만들어 더 나은 사용자 경험을 제공할 예정입니다.
3. 예외처리 기능 강화: 현재 구현된 기능들을 테스트할때 예측하지 못한 예외가 발생하는 문제가 있습니다. 백엔드 및 프론트엔드의 예외처리 코드를 보완하여 유저의 서비스 사용 과정에서 처리되지 못한 에러가 발생하는 것을 최대한 막고, 사용자 경험을 개선할 예정입니다.

4 향후 추진계획

4.1 향후 계획의 세부 내용

4.1.1 알림기능

SSE(서버 전송 이벤트)는 서버가 HTTP를 통해 웹 클라이언트에 실시간 업데이트를 푸시할 수 있도록 하는 웹 기술입니다. 클라이언트 측에서 지속적으로 폴링할 필요 없이 사용자에게 실시간 알림을 전달하는 가볍고 효율적인 방법을 제공합니다. 이를 통해 향상된 사용자 경험을 기대할 수 있습니다. 실시간 알림을 통해 사용자의 참여를 유도하고 커뮤니티 활동에 대한 정보를 받아 더욱 몰입도 높은 경험을 제공합니다.

4.1.2 Terraform을 통한 배포

IaC(Infrastructure as Code) 도구 중 하나인 Terraform을 활용하면 AWS 리소스의 프로비저닝 및 관리를 자동화 할 수 있습니다. 이를 통해 배포 프로세스의 일관성, 확장성을 기대할 수 있습니다. 내부적으로는 ECS와 Fargate를 사용해 애플리케이션 가용성 및 안정성을 보장하고 컨테이너 환경에서 안정적인 운영을 기대할 수 있습니다. 마지막으로 CI/CD 도구인 Github Action과 연계를 통해 변경된 내용을 즉시 배포해 확인할 수 있도록 할 것입니다.

5 고충 및 건의사항

매주 지도 교수님과의 피드백 시간을 통해 프로젝트 추가 목표에 대해 고민했습니다. 도입을 고민했던 항목은 아래와 같습니다.

- a. ai 모델을 활용한 비속어 필터링 기능 구현
- b. 홍보를 통해 실사용자를 모집하여 베타테스트 진행

다만 현재 프로젝트 진행상황 및 일정 등을 고려할 때 둘 모두를 실행하는 것은 우리가 있겠다 판단하였기 때문에 팀원간 논의를 진행하였고, 이 과정에서 홍보 및 베타테스트를 진행하는 것이 보다 실용적인 서비스를 만드는데에 도움이 되겠다는 것으로 의견이 모여 베타 테스트를 도입하는 것을 생각하고 있습니다.

하지만 저희 프로젝트 특성상 같은 학교 학생보다도 같은 학과를 중심으로 테스트 유저를 모집하고 진행해야 하기에 홍보를 진행하는 방법 마련에 있어 어려움이 있습니다. 해당 부분에 대해 많은 조언과 도움 부탁드립니다.

3.2 1차 중간보고서 발표자료

Decl

국민대학교 소프트웨어융합대학 소프트웨어전공
2024. 04. 05

다학제간 캡스톤 디자인 중간발표

7조 Decl팀

20190155 권지아
20171659 윤용배
20212674 김동윤
20213032 윤홍현

목차

CONTENTS

01 프로젝트 기획

프로젝트 Decl의 기획 과정

02 주요 기능 소개

Decl의 주요 기능

03 진행 상황

현재까지의 프로젝트 진행상황

04 향후 개발 계획

앞으로의 프로젝트 발전 및
개발 계획

73 Decl

프로젝트 기획

이 프로젝트 기획

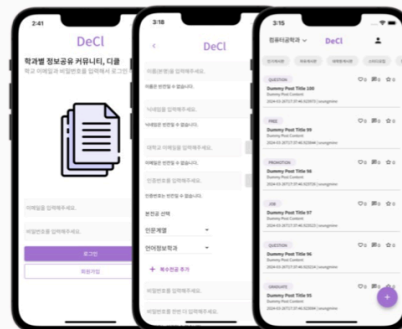
프로젝트 소개

학과별 정보 공유 커뮤니티, 디클

디클은 전국의 대학생들이 학과별로 모여 정보를 공유할 수 있는 커뮤니티 서비스입니다.

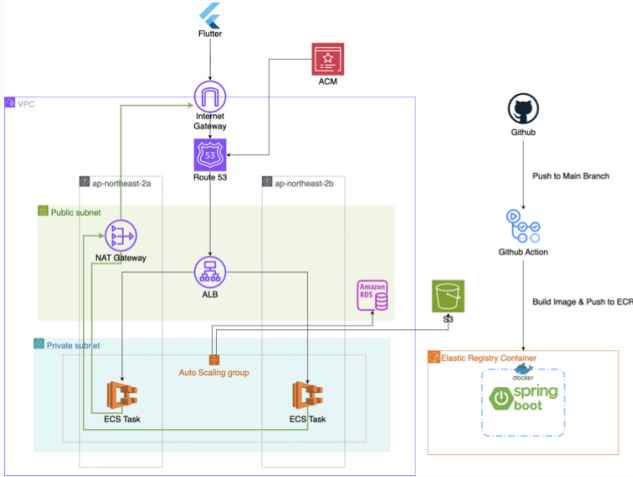
전공을 공통 관심사로 가진 대학생들이 모여 좀 더 깊은 관심을 공유할 수 있는 소통의 창구가 되고자 합니다.

Frontend: Flutter 3.13.0, Dart 3.1.0
Backend : Spring Boot 3.2.3, Kotlin 1.9.0, Kotlin-jdsl 3.3.1
Infrastructure : Terraform, AWS ECS + Fargate
CI/CD : github action



프로젝트 기획

이 프로젝트 기획



Frontend: Flutter 3.13.0, Dart 3.1.0
 Backend: Spring Boot 3.2.3, Kotlin 1.9.0, Kotlin-jdsl 3.3.1
 Infrastructure: Terraform, AWS ECS + Fargate
 CI/CD: github action

시장 포지셔닝

이 프로젝트 기획



x: 유저들이 모이는 기준
 y: 같은 공통 관심사를 가진 유저 간의 소통

심층 인터뷰

이 프로젝트 기획

- 심층인터뷰: 사용자의 잠재적인 혹은 구체적인 니즈를 확인하기 위해 진행하는 리서치.



충북 22세 여, 권00씨 (식품영양학과)

"저희 학교의 식품영양학과는 소수과라 에브리타임 내의 학과별 게시판이 활성화 되어있지 않기도 하고 사용하기 부담스러워요."



서울 25세 여, 신00씨 (통계학과)

"대학원에 진학할 예정이라 같은 통계 분야에 있는 다양한 학교 학생들과 더 많은 정보를 교류하고 싶어요."



전북 26세 남, 서00씨 (농업경제학과)

"전국에 개설된 대학이 많지 않은 학과라 다른 학교 학우들과의 교류가 필요해요. 같은 과 끼리만 이해할 수 있는 주제들이 있어요."

2024. 03. 27 자체 IDI 조사: n=3

심층 인터뷰

이 프로젝트 기획

⇒ 같은 전공을 가진 다양한 학교의 학생들과 한 곳에 모여 소통하고 싶은 대학생들의 니즈가 있다!

주요 기능 소개

02 기능 소개

학교 이메일 인증을 통한 회원가입 및 로그인

전국 40개 대학교의 이메일 SUFFIX를 데이터베이스에 저장해 회원가입시 검증 및 이메일 인증을 진행하고 있습니다.

다중 전공 선택 기능

회원가입시 최대 2개의 학과를 선택, 메인 탭에서 전환하여 본전공과 복수전공의 정보를 모두 챙길 수 있습니다. (주전공 및 복수전공 변경은 90일마다 가능)

카테고리 필터링 및 검색 기능

같은 전공 분야의 학우들과 나누고 싶은 관심사에 따라 카테고리를 분류하고 필터링 기능을 제공합니다. 또한 원하는 검색어를 통해 글을 찾을 수 있도록 검색 기능을 제공합니다.

게시글 CRUD, 마이페이지 기능 등

이외에 커뮤니티 이용을 위한 게시글 및 댓글의 CRUD, 마이페이지에서 확인이 가능한 작성글 조회, 댓글 단 글 조회, 스크랩 글 조회, 회원정보 변경 등의 추가 기능을 제공합니다.

진행 상황

03 진행 상황

학교 이메일 인증을 통한 회원가입 및 로그인 (진행도 100%)

[완료]

- 전국 40개 대학교의 이메일 SUFFIX를 데이터베이스에 저장해 회원가입시 검증 및 이메일 인증을 진행하도록 구현.
- 인증된 학교 이메일을 통한 회원가입, 로그인 기능 구현.
- JWT토큰 만료 여부에 따른 클라이언트 로그인 상태 관리 (REFRESH TOKEN 토큰 만료시 자동 로그아웃) 구현.



진행 상황

03 진행 상황

다중 전공 선택 기능 (진행도 70%)

[완료]

- 회원가입시 유저의 선택에 따라 최대 2개의 전공 입력이 가능하도록 구현.
- 유저가 게시판의 메인 페이지에서 전공을 전환하여 본인의 복수전공 학과 게시글을 확인할 수 있도록 구현.

[예정]

- 회원정보 변경 페이지에서 90일마다 전공 자체를 변경할 수 있도록 학과 수정 기능 구현.



진행 상황

03 진행 상황

카테고리 필터링 및 검색 기능 (진행도 70%)

[완료]

- 자유 전공게시판, 취준 게시판, 대학원 게시판, 홍보 게시판 등 전공이 같은 학생들에게 필요한 카테고리 기능 구현

[예정]

- 유저가 원하는 정보를 검색할 수 있도록 검색기능 구현.



진행 상황

03 진행 상황

게시글 CRUD 및 마이페이지 기능 (진행도 50%)

[완료]

- 메인페이지에서 게시글 조회 및 커서 기반 페이지네이션 구현
- 이미지 첨부을 포함한 게시글 작성 기능 구현
- 게시글 상세 페이지 및 댓글 조회 기능 구현
- 마이페이지 메인 스크린 구현

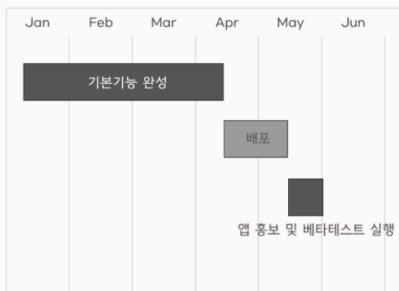
[예정]

- 게시글 삭제 및 수정 기능 구현
- 게시글 상세페이지 댓글 페이지네이션 구현
- 마이페이지 상세 스크린 구현 (내가 작성한 글, 댓글 단 글, 스크랩한 글)
- 마이페이지 회원정보 수정 기능 구현
- 실시간 댓글 알림 기능 구현



향후 개발 계획

04 향후 개발계획



향후 계획

- 4월 중순까지 기본기능 개발 완료
- 4월 중순~5월 첫째주까지 서버 배포 및 앱스토어/플레이스토어에 배포 (예상치 못한 reject 발생시 지연가능성 O)
- 5월 첫째주부터 약 2주간 베타테스트 진행. 특정 학과를 정하여 진행할 예정

* 개발 상황에 따라 변동 가능성이 있습니다.

**7조 발표를 마칩니다.
들어주셔서 감사합니다.**

발표자 권지아

제 4 장

결과보고서

4.1 결과보고서

4.2 결과보고서 발표자료

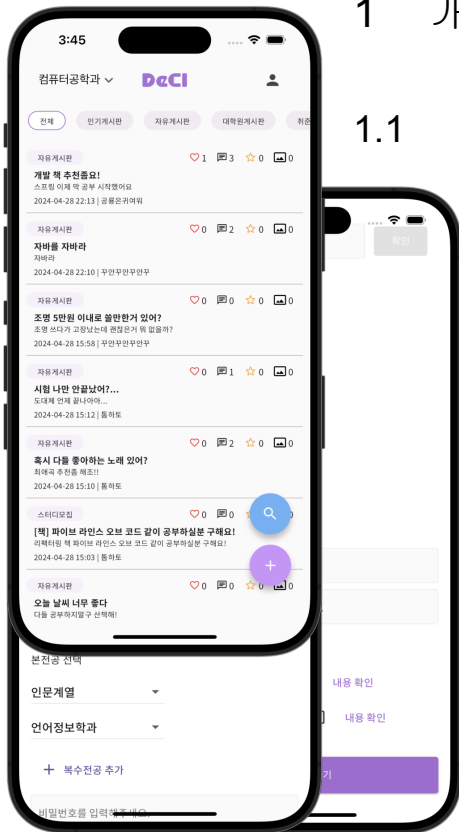
4.1 결과보고서

목 차

1	개요	4
1.1	프로젝트 개요	4
1.2	추진 배경 및 필요성	4
2	개발 내용 및 결과물	5
2.1	목표	5
2.2	연구/개발 내용 및 결과물	6
2.2.1	연구/개발 내용	6
2.2.2	시스템 기능 및 구조 설계도	6
2.2.3	활용/개발된 기술	6
2.2.4	현실적 제한 요소 및 그 해결 방안	6
2.2.5	결과물 목록	6

2.3	기대효과 및 활용방안	6
3	자기평가	7
4	참고 문헌	7
5	부록	7
5.1	사용자 매뉴얼	7
5.2	운영자 매뉴얼	7
5.3	배포 가이드	7
5.4	디클 매뉴얼	7
5.5	디클에 대한 기술 문서	7

1 개요



1.1 프로젝트 개요

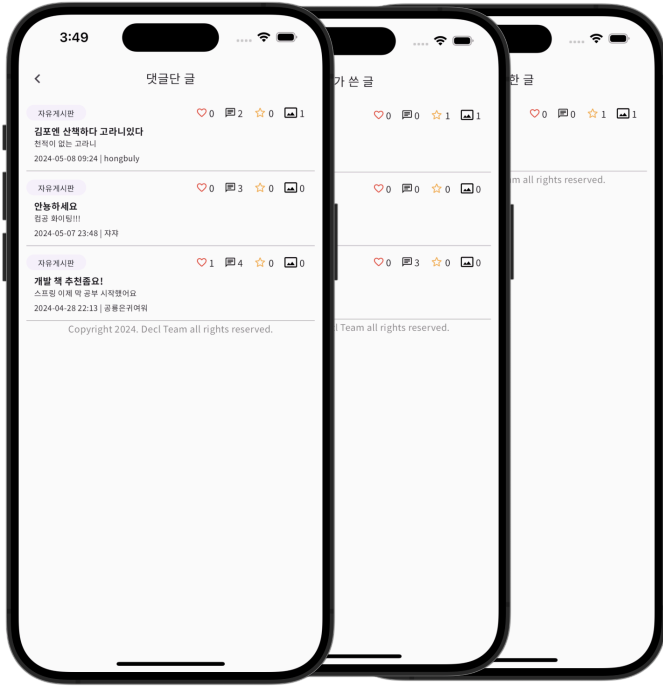
회원가입 기능

- 사용자는 전국 대학교 이메일 주소를 사용하여 가입할 수 있다. 이메일 주소는 데이터베이스에 사전 등록되어 있으며, 등록되지 않은 이메일 주소로는 가입이 불가능하다.
- 사용자는 가입 과정에서 본인의 전공 및 부/복수 전공을 선택할 수 있다. 총 두 개의 전공 정보를 입력할 수 있다.



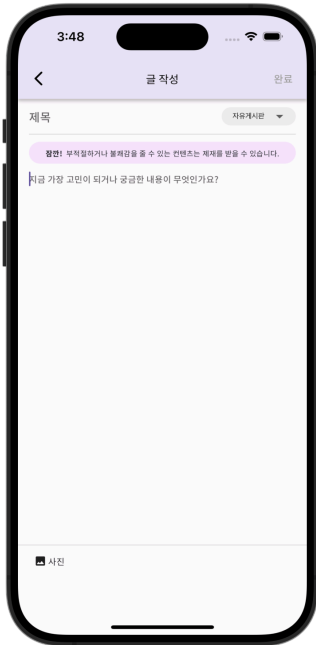
로그인 기능

- 사용자는 JWT 토큰을 사용하여 로그인한다. 로그인 시 발급되는 Access Token과 Refresh Token을 사용하여 사용자 인증을 관리한다.
- Access Token이 만료되었을 경우, Refresh Token을 사용하여 새로운 Access Token을 재발급 받을 수 있습니다.



글 목록 불러오기 기능

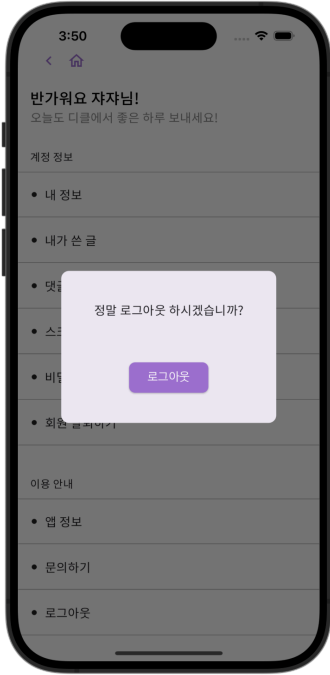
- 사용자는 현재 활성화된 학과 기준 및 커뮤니티 기준으로 작성된 글을 볼 수 있다.
- 자유게시판, 대학원 게시판 등 총 5개의 게시판과 전체 게시판, 좋아요 개수가 10개 이상인 글을 모아보는 인기 게시판이 있다.



글 작성 기능

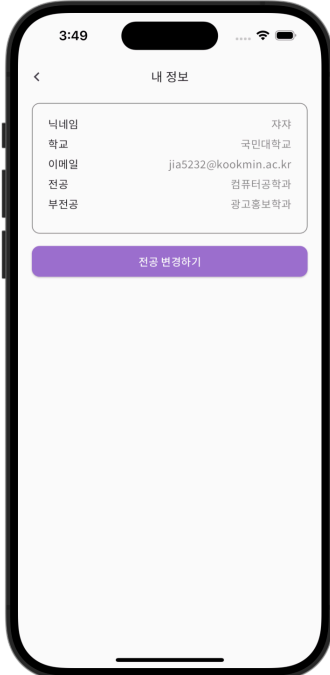
- 사용자는 특정 커뮤니티를 선택하여 글을 작성할 수 있다. 글 작성 시, 제목, 내용, 관련 이미지 등을 추가할 수 있다.

- 로그인된 사용자는 마이페이지를 통해 내 정보 확인, 내가 쓴 글, 댓글 단 글 확인, 로그아웃 등의 기능을 이용할 수 있다.



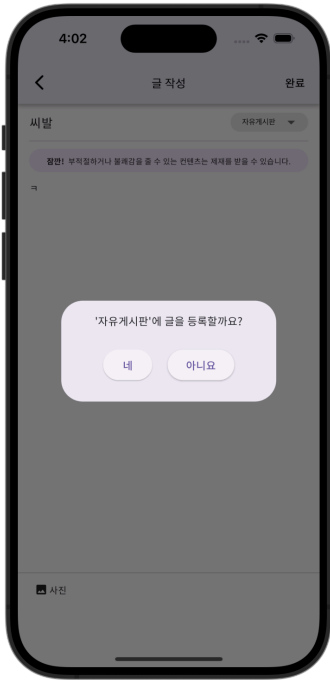
내 정보

- 사용자는 자신의 닉네임, 학교, 전공 등의 정보를 확인할 수 있으며, 전공 정보를 변경할 수 있는 기능을 제공한다



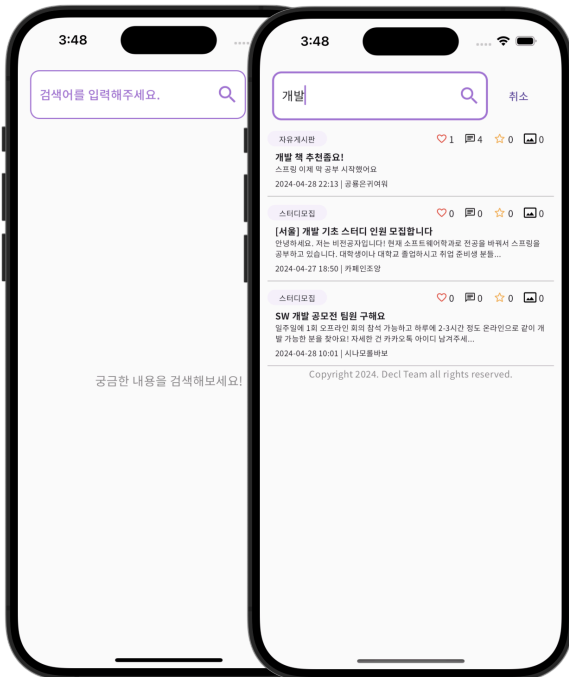
내가 쓴 글, 댓글 단 글, 스크랩 한 글 페이지

- 사용자는 자신이 작성한 글, 댓글을 단 글, 스크랩한 글을 한눈에 볼 수 있는 기능을 제공받는다.



검색 페이지

- 사용자는 특정 키워드를 제목이나 내용에 포함한 글들을 검색할 수 있다.

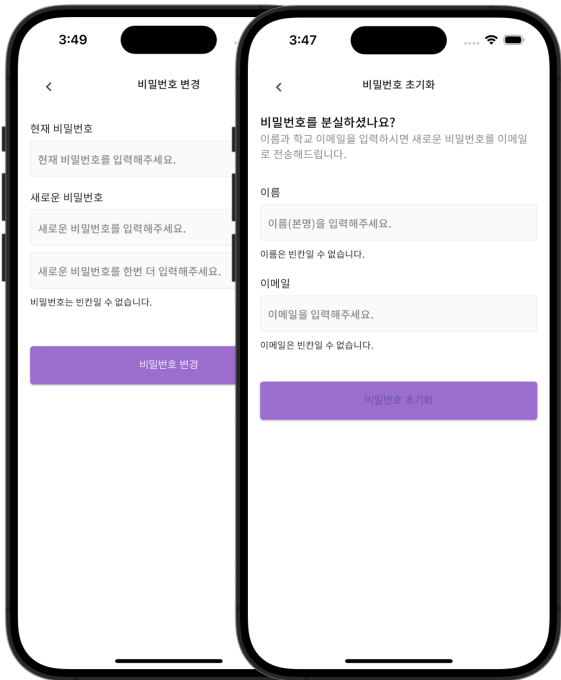


로그아웃 기능



- 사용자는 언제든지 로그아웃을 할 수 있으며, 로그아웃 시 모든 세션이 종료된다.

비밀번호 변경 기능



- 사용자는 현재 비밀번호를 확인한 후 새로운 비밀번호로 변경할 수 있다.

비밀번호 초기화 기능

- 비밀번호를 분실한 경우, 사용자는 등록된 이메일을 통해 새로운 랜덤 비밀번호를 발급받고 초기화할 수 있다.

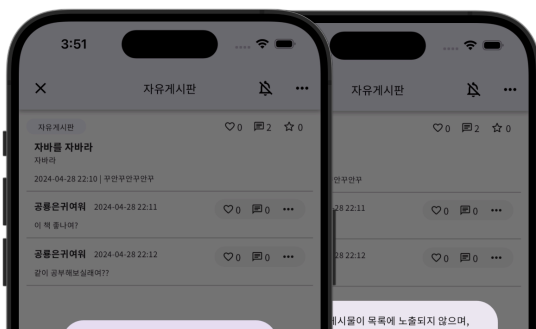
비속어 필터링

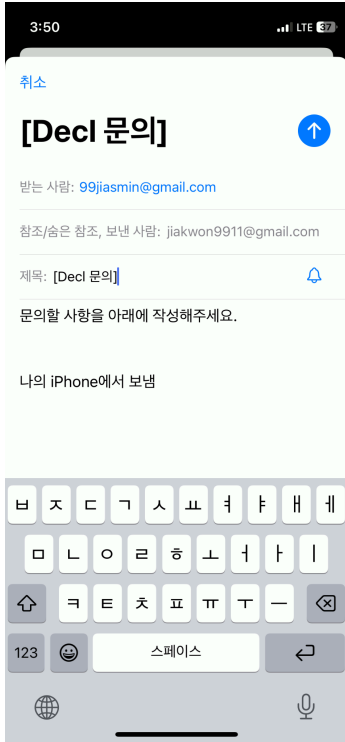
- AI 모델을 학습시켜 사용자가 작성하는 글에 비속어가 포함되어 있을 경우, 경고 메시지를 표시한다.



신고 기능

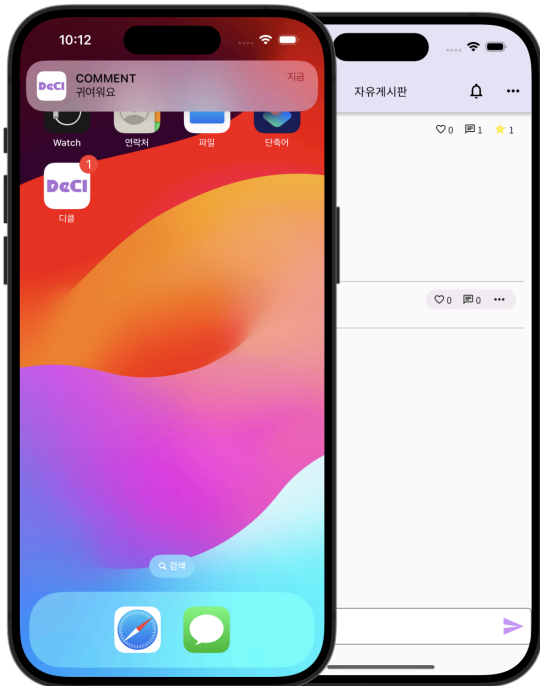
- 사용자는 특정 유저의 글, 댓글, 대댓글이 커뮤니티 가이드라인을 위배한다고 판단될 경우, 신고할 수 있다. 신고가 10회 누적된 유저는 자동으로 정지된다.





차단 기능

- 사용자는 특정 유저를 차단할 수 있으며, 차단된 유저의 글, 댓글, 대댓글은 보이지 않는다.

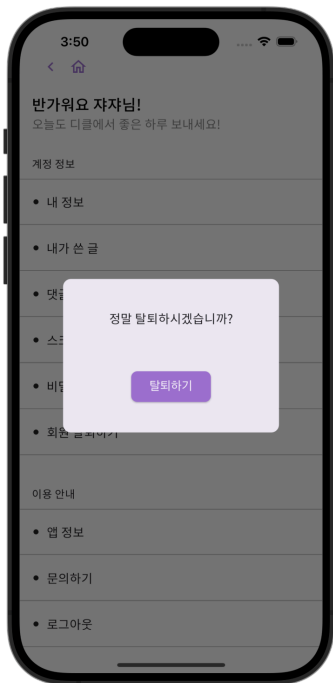


실시간 댓글 알림

- 사용자는 자신이 작성한 글 또는 댓글에 대댓글이 달리는 경우 실시간으로 알림을 받는다. 알림을 클릭하면 해당 글로 이동할 수 있다.

문의

- 사용자는 고객센터 메일로 문의 사항을 작성할 수 있으며, 문의 사항에 대한 답변을 이메일로 받을 수 있다.



회원 탈퇴

- 사용자는 자신의 계정을 삭제할 수 있는 기능을 제공받는다.

1.2 추진 배경 및 필요성

대학 생활은 단순히 학문적 지식을 쌓는 곳을 넘어, 다양한 사람들과의 소통과 협업을 통해 사회에 첫발을 내딛기 위한 중요한 과정이다. 이 과정에서 학과나 전공별 소통은 학생들이 직면한 공통의 고민을 해결하고, 필요한 정보를 공유하는 데 있어 필수적인 역할을 한다. 하지만 현실은 대학생들이 각 학과나 전공별로 소통할 수 있는 창구가 매우 부족하다는 것이다. 많은 학생들이 같은 학교 내의 사람들과의 소통에 부담을 느끼거나, 그 한계를 경험하고 있다.

기존에 존재하는 서비스들, 예를 들어 에브리타임은 학교별로 소통하는 기능을 제공하지만, 실질적으로 학과나 전공별 깊이 있는 소통에는 한계가 있다. 이러한 서비스들은 대체로 학교 생활 전반에 대한 정보를 공유하는 데 초점을 맞추고 있으며, 특정 학과나 전공에 대한 세부적인 정보나 고민 상담 등을 지원하는 데는 부족함이 있다.

이러한 상황에서, 전국의 대학생들이 자신의 학과나 전공에 대해 자유롭게 정보를 주고받고, 고민이나 경험을 공유할 수 있는 새로운 플랫폼의 필요하다고 생각되었다.

따라서, 우리는 전국의 대학생들이 사용할 수 있는 학과별 커뮤니티 어플리케이션을 개발하기로 결정했다. 이 어플리케이션은 사용자가 자신의 학과를 선택하여 가입함으로써, 같은 학과나 전공을 가진 학생들과의 소통을 가능하게 한다. 또한, 학과별 맞춤 정보 공유, 스터디 그룹 형성, 취업 정보 공유, 그리고 학과 관련 고민 상담 등 다양한 기능을 제공하여, 학생들이 학과 생활의 질을 향상시킬 수 있도록 지원한다.

이러한 서비스는 학생들에게 더욱 풍부하고 다양한 학과 생활을 경험할 수 있는 기회를 제공함으로써, 학과 내 소통의 폭을 넓히고, 학생들 간의 유대감을 강화하는 데 기여할 것이다. 최종적으로는 학과 커뮤니티를 강화하고, 대학 생활 전반에 긍정적인 영향을 미치는 것을 목표로 하고 있다.

2 개발 내용 및 결과물

2.1 목표

디클은 전국 대학생들이 학과를 중심으로 모여 소통할 수 있는 학과별 커뮤니티 어플이다. 기존 에브리타임과 같은 커뮤니티는 소속 학교를 중심으로 모이므로, 같은 전공 분야에 있는 다양한 학교의 학생들과는 소통이 어렵다는 문제가 있다. 또한 에브리타임 내의 학과별 게시판도 활성화 되어있는 일부 학과들 외에는 사실상 사용하는 사람이 적다. 따라서 전국의 모든 대학교 재학생들이 학과별로 모여 정보를 공유하고 자유롭게 소통할 수 있는 커뮤니티를 개발하고자 한다.

2.2 연구/개발 내용 및 결과물

2.2.1 연구/개발 내용

1. 학교 이메일 인증을 통한 회원가입 및 로그인

전국 401개 대학교의 이메일을 수집하여 SUFFIX를 데이터베이스에 저장했다. 따라서 이메일을 통해 회원가입 시, 데이터베이스에서 조회가 가능한 401개 대학에 포함되어 있다면 정상적으로 인증 메일을 발송해 회원가입 절차를 따르게 된다.

2. 전공 선택

커리어넷(<https://www.career.go.kr/cnet/front/base/major/FunivMajorList.do>) 에서 조회 가능한 학과들을 디클 서비스에 활용한다.

3. 전공 관심사에 따른 카테고리 분류

게시글 객체를 데이터베이스에 저장할 때, 카테고리 필드를 추가하여 쿼리를 통해 원하는 카테고리의 게시글을 검색하도록 구현했다.

4. 게시글 CRUD, 마이페이지

스프링 부트 및 JPA를 활용해 데이터베이스와 객체간 매핑 문제를 해결했고 Controller-Service-Repository 레이어로 나누어 CRUD 및 마이페이지 기능을 구현했다.

5. 알림 기능(추가)

SSE 기술을 활용해 사용자가 접속 시 실시간 알림을 받을 수 있도록 한다.

2.2.2 시스템 기능 요구사항

1. 회원가입 관련(완료)

- 1.1. 학교 이메일 인증 기능(완료)
- 1.2. 회원가입 기능(완료)
- 1.3. JWT 기반 API 요청(완료)
- 1.4. 아이디 및 비밀번호 찾기(완료)

2. 게시물 CRUD(완료)

- 1.1. 게시물 페이지네이션 조회(완료)
- 1.2. 게시물 작성 및 이미지 업로드(완료)
- 1.3. 게시물 삭제(완료)
- 1.4. 게시물 수정 및 이미지 수정(완료)
- 1.5. 게시물 검색 기능(완료)

3. 게시물 상세(완료)

- 3.1. 댓글 CRUD(완료)
- 3.2. 대댓글 CRUD(완료)
- 3.3. 게시물 좋아요 기능(완료)
- 3.4 스크랩 기능(완료)

4. 마이 페이지(완료)

- 4.1. 내가 작성한 글 조회(완료)
- 4.2. 내가 댓글단 글 조회(완료)
- 4.3. 내가 스크랩한 글 조회(완료)
- 4.4. 전공 변경 기능(완료)
- 4.5. 비밀번호 변경(완료)
- 4.6. 회원 탈퇴(완료)

5. 알림 기능(완료)

6. 신고 기능(완료)

- 6.1. 신고 누적 시 게시물, 댓글, 대댓글 차단 기능(완료)
- 6.2. 신고된 게시물, 댓글, 대댓글이 있는 사용자 Create 기능 일시정지(완료)

7. 차단 기능(완료)

2.2.3 시스템 비기능(품질) 요구사항

1. 고가용성(달성)

AWS 두 가용영역(ap-northeast-2a, ap-northeast-2b)에 배포한 이미지를 ECS Task로 실행하여 고가용성을 달성함

2. 확장성(달성)

컨테이너의 CPU와 Memory 사용률이 70%를 넘어갈 시, Autoscaling으로 추가적인 컨테이너가 실행되어 확장성을 달성함

3. 보안(달성)

- 1) 로직 서버를 AWS private subnet에 배치하여 외부에서 로직 서버에 함부로 접근할 수 없게 함.
- 2) 사용자 암호를 해시하여 데이터베이스에 저장함.
- 3) JWT 토큰을 사용해 사용자 인증 메커니즘을 구현함.
- 4) HTTPS 설정을 통해 보안성을 높임.

4. 성능(달성)

- 1) 게시글을 불러오는 API 호출 시, 페이지네이션 및 batch size 설정으로 발생하는 쿼리 횟수를 줄여 성능 최적화를 달성함.
- 2) 코틀린 코루틴을 활용해 응답 시간을 최적화함.

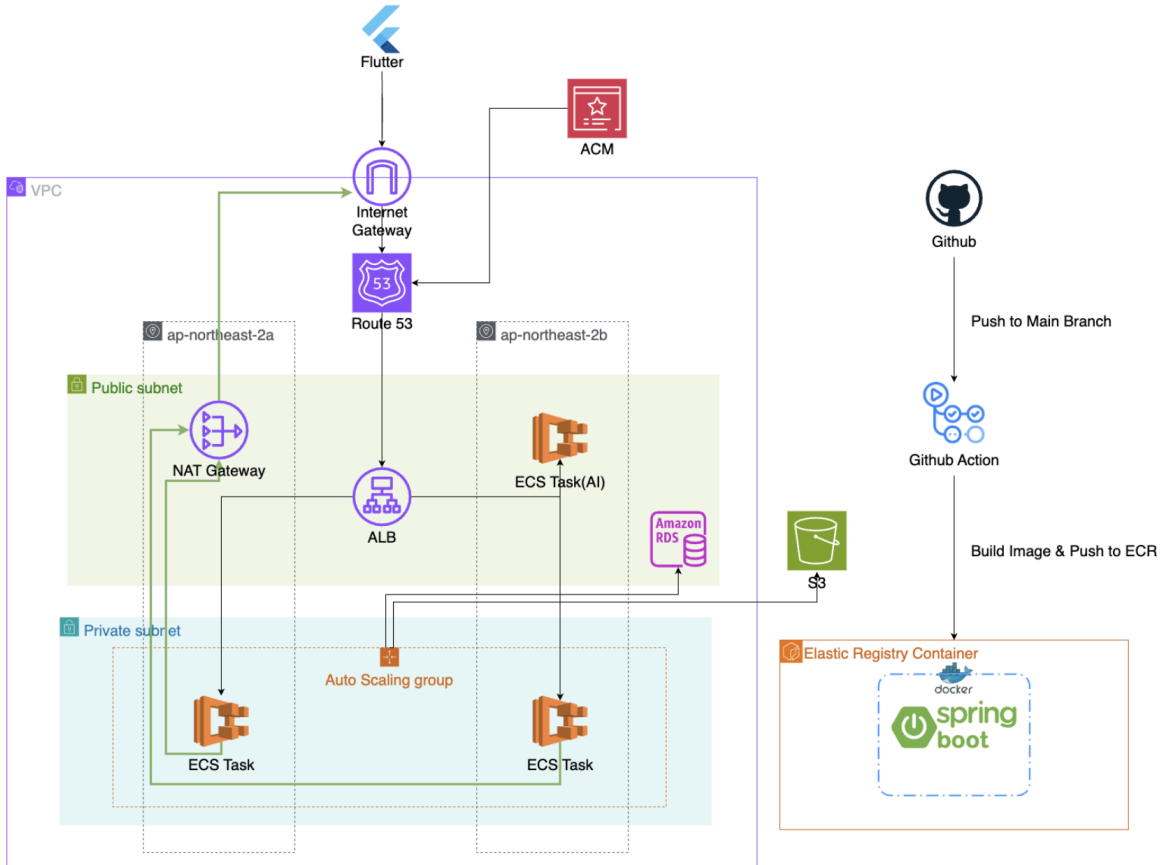
5. 유지관리(달성)

- 1) 프론트엔드와 백엔드 협업 시, Swagger를 통해 API 문서 자동화를 달성함.
- 2) 발생하는 API에 대해 AOP를 적용해 로그를 확인하고 AWS CloudWatch를 통해 조회 가능함.
- 3) github action을 활용해 CI/CD를 구축함

6. 품질(달성)

- 1) 테스트 코드 작성을 통해 코드 품질을 높임.
- 2) 코드 리뷰 및 리팩토링을 통해 코드 가독성을 높임.

2.2.4 시스템 구조 및 설계도



보안성을 고려하기 위해 VPC를 public subnet과 private subnet으로 구분했다. public subnet에는 private subnet의 트래픽을 처리할 수 있는 NAT Gateway와 사용자의 데이터를 저장하기 위한 데이터베이스를 배치했다. 사용자로부터 요청이 발생하면 로드밸런서에 정의한 Listener Rule에 따라 적절한 Target Group으로 트래픽을 포워딩 한다. 마지막으로 HA와 Scalability를 고려하기 위해 두 개의 가용영역에 Autoscale 기능을 활용하여 각각 ECS Task를 실행한다.

2.2.5 활용/개발된 기술

AI



Python



Scikit learn

Frontend



Dart



Flutter



Riverpod



Dio

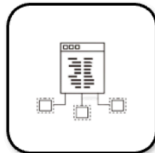
Backend



Kotlin



Spring Boot



JPA



LINE

Kotlin jdsl



Swagger



MySQL

Infrastructure



Terraform



ECS



Fargate



S3



Docker



Github action

AI

비속어 필터링 기능 구현을 위해 파이썬의 사이킷런 라이브러리를 활용했다. 욕설 유무에 해당하는 y값이 0,1 임을 고려해 로지스틱 모델로 기능을 구현했다.

Frontend

크로스 플랫폼 앱 개발 플랫폼으로 플러터를 채택했다. 이를 통해 단일한 코드 베이스를 바탕으로 Android, iOS뿐 아니라 PC와 웹 호환성까지 고려했다. 상태관리 라이브러리는 Riverpod, 백엔드와 통신을 위해 Dio 라이브러리를 채택했다.

Backend

객체지향성을 고려하고 다양한 라이브러리를 지원하는 스프링 부트를 프레임워크로 채택했다. 개발 언어는 Java와 Kotlin 중 Null safety, 함수형 프로그래밍 지원 및 간결성과 가독성 측면에서 우수한 Kotlin을 채택했다. 데이터베이스와 객체간 매핑 문제를 해결하기 위해 JPA를 사용했고 동적 쿼리 작성을 위해 Kotlin JDSL을 사용했다. QueryDSL은 동적 쿼리 작성을 위해 메타데이터를 사용해야하는데 해당 라이브러리는 메타데이터를 생성하지 않고도 동적 쿼리를 작성할 수 있다.

Infrastructure

배포를 위해 가장 많이 사용하는 AWS를 채택했다. 수동으로 배포하는 대신, 클라우드 및 온프레미스 인프라를 코드로 관리할 수 있는 테라폼을 활용했다. 이를 통해 AWS 리소스 생성, 수정, 삭제를 코드 레벨에서 관리 가능한 이점을 챙겼다. 빌드한 이미지를 실행할 도구로 서버리스 기반 Fargate를 사용했고 container orchestration 도구는 ECS를 사용했다. EKS가 가지는 여러 이점이 존재하지만 비용 및 오버 엔지니어링이라 판단했기 때문이다. 게시글과 함께 업로드하는 이미지는 S3, 도커 이미지 저장소는 ECR Private Registry를 사용했다. 마지막으로 CI/CD를 위해 github action를 사용했는데 이미 github 기반으로 협업하므로 추가적인 외부 플랫폼의 도움을 받지 않아도 되는 장점이 가장 크게 작용했기 때문이다.

2.2.6 현실적 제한 요소 및 그 해결 방안

비용

좋은 아키텍처를 사용할수록 그만큼의 비용을 지불해야한다. 서버 비용은 매달 15만원으로 추산했고 본격적으로 배포한 4월, 5월의 서버비용은 최대 30만원정도로 예상했다. 학교에서 캡스톤디자인을 위해 인당 10만원씩 총 40만원의 지원을 해주어 해당 금액으로 서버 비용 및 로고 비용을 충당했다.

2.2.7 결과물 목록

구글 플레이스토어



앱스토어 (ios)



2.3 기대효과 및 활용방안

- 1) 전공 관련 정보 공유 강화
학과별로 모인 유저들이 전공 관련 정보를 쉽게 공유할 수 있어, 학업에 도움이 되는 정보를 쉽게 얻을 수 있다.
- 2) 네트워킹 및 협업 기회 제공
같은 학과 출신의 사용자끼리의 소통은 졸업 후의 진로 결정이나 협업 기회를 제공한다.
- 3) 자기 계발 및 취미 활동 공유
학과 이외의 관심사나 취미에 관한 정보 또한 공유하고 함께 활동할 수 있는 기회를 제공한다

3 자기평가

자 체 평 가 서

팀명	7조	성명	윤홍현
프로젝트제목	디클 (Decl)		
자체 평가	<p>프로젝트 수행 소감:</p> <p>하이브리드 앱 개발을 위해 Flutter를 배운 것은 새로운 기술을 배우는 것에 대한 두려움을 줄이는 데 도움이 되었습니다. 앱 개발에서 게시글 상세페이지, 게시글 등록페이지를 담당하여 개발하였습니다. 게시글 상세페이지에서는 댓글/대댓글 페이지네이션과 댓글, 대댓글 달기, 수정, 삭제를 구현하고, 신고하기/차단하기 등 백엔드와 연결하여 필요한 기능들을 구현하였습니다.</p> <p>평소에 팀프로젝트를 하면, 적극적이지 않고 자신이 해야 할 일들을 다 하지 못하는 팀원들과 프로젝트를 진행하여, 거의 대부분을 제가 프로젝트를 했기에, 팀으로 일하는 것을 싫어했습니다. 하지만 이번 캡스톤에서는 저에게 필요한 점들을 짚어주고, 도움을 받을 수 있는 팀원이 함께 하여, 기술을 빠르게 습득하고 개발을 하는데 도움이 되었습니다. 이러한 경험을 통해 같이 일하는 팀원의 책임감이나 능력이 중요하다는 점을 다시한번 느꼈습니다. 저에게 이 점을 적용하여 함께 일하고 싶고, 함께 성장하고 싶은 팀원이 될 수 있도록 노력해야 겠다고 느꼈습니다.</p> <p>개선요구사항:</p> <p>프로젝트를 하면서, 저 자신에게 개선해야겠다고 느낀 점은 깃 사용이었습니다. 오류를 만났을 때 수정을 해야하는데, 깃 커밋을 쪼개지 않고 계속 작업을 하는 버릇이 있었습니다. 앞으로 이러한 습관을 개선해야 겠다고 느꼈습니다.</p> <p>프로젝트에서 개선할 점으로는 코드를 간략하게 하고, 로딩시간을 줄이기 위한 방법을 더 모색해야 겠다고 느꼈습니다.</p> <p>발전적 제언:</p> <p>하이브리드 앱 개발 말고 네이티브 앱 개발에서만 할 수 있는 것들이 있기 때문에, 네이티브 앱 개발 중 IOS에 대해 더 공부해 봐야겠다고 느꼈습니다.</p> <p>프로젝트에서 발전해야 할 점으로는 대학교 어플이니까 시간표와 식사표가 같이 있는것이 사용자들이 더 앱을 자주 사용하게 만들 수 있는 기능이기에 때문에 추가적으로 있으면 좋겠다고 생각했습니다.</p>		

자 체 평 가 서

팀명	7조	성명	윤웅배
프로젝트제목	디클		
자체 평가	<p>프로젝트 수행 소감:</p> <p>이전 프로젝트를 수행하며 개선하거나 새로 적용해보고 싶은 기술들이 여럿 있었습니다. 대표적으로 Kotlin + Spring, CI/CD, 도메인 주도 개발, 테스트 커버리지, Terraform 활용뿐만 아니라 브랜치 전략과 깃허브를 활용한 협업을 경험해보고 싶었습니다. 캡스톤 프로젝트에서 그 기회가 닿아 새로운 기술들을 적용해볼 수 있었습니다. 피상적으로 알고만 있던 지식들을 실제 코드로 구현하는 과정에서 미처 알지 못했던 여러 문제들을 겪었고 이를 해결하는 과정에서 성장할 수 있어 좋았습니다. 특히 테라폼을 활용해 AWS에 배포하는 경험을 통해 AWS에서 제공하는 서비스에 대해 많은 공부를 할 수 있어 좋았습니다. 개발부터 앱스토어에 배포하는 전체 프로세스를 체험하고 나아가 이를 유지보수하며 실제 현업에 가까운 경험을 할 수 있는 소중한 기회라고 생각합니다. 개발 외적으로도 좋은 팀원들과 의견 공유와 토의하는 과정에서도 개인적 성장을 이루었다 생각합니다.</p> <p>개선요구사항:</p> <p>모든 부분에서 클린 코드 원칙을 지키며 코드를 작성하고 싶었지만 더 나은 코드 작성 방법을 알지 못해 어쩔 수 없이 원칙을 어겼던 적이 있습니다. 기회가 된다면 이런 부분도 리팩토링 하고 싶습니다.</p> <p>발전적 제언:</p> <p>협업 프로세스를 개선해 매주 회의를 갖는 대신 매일 짧은 회의를 가져 모든 팀원이 동일한 정보를 팔로우업 할 수 있도록 한다.</p>		

자 체 평 가 서

팀명	Decl	성명	김동윤
프로젝트제목	Decl		
자체 평가	<p>프로젝트 수행 소감:</p> <p>이번 프로젝트는 제가 백엔드 개발자로 참여하는 두 번째 프로젝트였습니다. 백엔드 개발에 대한 이해도가 여전히 부족하다고 느끼는 가운데, 이번 프로젝트를 통해 처음으로 다루어 보는 프레임워크와 프로그래밍 언어에 도전했습니다. 프로젝트 시작 전, 저는 새로운 기술 습득에 많은 시간을 할애했습니다. 이 과정에서 기본 개념을 익히는 것뿐만 아니라, 다양한 개발 도구와 라이브러리에 대한 이해도를 높이기 위해 노력했습니다.</p> <p>프로젝트를 진행하면서, 같은 백엔드 포지션에 있는 팀원으로부터 많은 것을 배울 수 있었습니다. 그들의 경험과 지식 덕분에 개발 과정 중 발생할 수 있는 다양한 문제들을 쉽게 해결할 수 있었습니다. 특히, 새로운 프레임워크와 언어를 사용함에도 불구하고, 프로젝트를 성공적으로 마무리할 수 있었던 것은 큰 성취감을 안겨주었습니다. 이 경험은 저에게 많은 자신감을 주었고, 앞으로의 개발자 생활에 긍정적인 영향을 미칠 것입니다.</p> <p>그러나, 프로젝트의 개발 부분 외에 인프라스트럭처 설계와 프로젝트 세팅에 직접 참여하지 못한 것은 아쉬움으로 남습니다. 이 부분은 프로젝트의 성공적인 진행을 위해 매우 중요한 요소임에도 불구하고, 저는 이번 기회에 그에 대한 경험을 충분히 쌓지 못했습니다. 하지만, 이러한 아쉬움도 팀원으로부터 관련 지식을 배울 수 있는 기회가 되었습니다. 이 경험을 바탕으로, 다음 프로젝트에서는 프로젝트 세팅과 인프라스트럭처 설계를 주도적으로 진행할 수 있는 능력을 갖추고자 합니다.</p> <p>또한, 이번 프로젝트를 통해 처음으로 앱스토어와 플레이스토어에 앱을 출시하는 경험을 할 수 있었습니다. 실제 사용자들이 저희 앱을 다운로드하고 사용하는 것을 보며, 개발자로서 큰 뿌듯함을 느꼈습니다. 이번 경험은 저에게 실제 사용자를 대상으로 한 애플리케이션 개발의 중요성과 그에 따른 책임감을 일깨워 주었습니다.</p> <p>앞으로 저는 더 많은 사용자를 확보하고, 높은 트래픽을 관리하며, 애플리케이션의 지속적인 유지보수를 진행하는 경험도 쌓고 싶습니다. 이번 프로젝트가 저에게 많은 것을 가르쳐 준 만큼, 다음 프로젝트에서는 이러한 경험을 바탕으로 더욱 성장한 모습을 보여주고 싶습니다. 프로젝트를 통해 배운 교훈과 경험은 제 개발자 생활의 소중한 자산이 될 것이며, 앞으로 더 좋은 개발자로 성장하는 데 큰 도움이 될 것입니다.</p> <p>개선요구사항:</p> <ol style="list-style-type: none"> 1. 인프라스트럭처 설계 지식 강화 및 실습: 프로젝트 초기 단계에서 인프라스트럭처 설계에 더 적극적으로 참여하고 싶습니다. 이를 위해 클라우드 서비스(AWS, GCP 등)에 		

대한 심화 학습을 진행하며, 인프라스트럭처를 구성하는 다양한 서비스들(예: 컴퓨팅, 스토리지, 네트워킹 등)의 조합과 최적화 방법을 습득할 계획입니다. 또한, 실제 프로젝트에서 인프라 설계를 담당하여, 이론과 실습을 병행하는 경험을 쌓고자 합니다.

2. **대용량 데이터 처리 최적화:** 대규모 트래픽과 대용량 데이터 처리는 현대 애플리케이션 개발에서 중요한 요소입니다. 이에 대응하기 위해, 데이터베이스 쿼리 최적화와 도메인 구조 개선에 주력할 예정입니다. **SQL** 쿼리 튜닝, 인덱싱 전략 수립, **NoSQL** 데이터베이스의 적절한 활용 등을 통해 데이터 조회 성능을 개선하고자 합니다. 또한, 캐싱, 데이터 파티셔닝, 비동기 처리 등의 기법을 도입하여 시스템의 전반적인 성능을 향상시킬 계획입니다.
3. **코드 품질 개선 및 리팩토링:** 코드의 중복을 최소화하고, 가독성과 유지보수성을 높이는 것은 지속 가능한 개발을 위해 필수적입니다. 이를 위해, 리팩토링에 좀 더 시간을 할애하며, 디자인 패턴, 함수형 프로그래밍 등의 원칙을 적용하여 더 깔끔하고 효율적인 코드를 작성할 계획입니다. 코드 리뷰 문화를 활성화하여 팀 내에서의 코드 품질 향상을 도모하고, 지속적인 학습을 통해 새로운 기술과 방법론을 접하며 개발 역량을 강화할 예정입니다.

발전적 제언:

1. **테스트 코드 작성 및 테스트 커버리지 향상:** 프로젝트의 안정성을 확보하기 위해 모든 기능에 대한 충분한 테스트 코드를 작성하는 것이 중요합니다. 이를 통해 코드 변경 시 발생할 수 있는 예기치 않은 오류를 사전에 방지하고, 높은 테스트 커버리지를 유지하여 신뢰성 있는 서비스를 제공할 수 있습니다. 테스트 주도 개발(**TDD**) 방식을 도입하여 개발 과정에서부터 테스트를 중심으로 코드를 작성하는 방법도 고려해볼 수 있습니다.
2. **주기적인 코드 리뷰 실시:** 코드 리뷰는 코드의 품질을 향상시키고, 개발자간의 지식 공유에도 매우 효과적인 방법입니다. 주기적으로 코드 리뷰를 실시하여 서로의 코드에 대한 구체적이고 건설적인 피드백을 제공함으로써, 개선된 코드 작성을 도모하고 개발자의 역량을 함께 향상시킬 수 있습니다. 특히, 신규 기능 개발이나 큰 변경 사항이 있는 경우에는 반드시 코드 리뷰를 통해 검증하는 과정을 거치는 것이 좋습니다.
3. **정기적인 진행 상황 공유 및 계획 토론:** 프로젝트의 진행 상황을 정기적으로 공유하고, 팀원 간의 의견을 나누며, 향후 계획에 대해 논의하는 것은 프로젝트를 계획대로 성공적으로 마치는 데 필수적입니다. 이러한 정기적인 회의를 통해 프로젝트의 목표 달성 상황을 점검하고, 문제가 발생했을 때 신속하게 대처할 수 있는 방안을 모색하며, 미래에 발생할 수 있는 잠재적 문제에 대해 사전에 대비할 수 있습니다.

자 체 평 가 서

팀명	7조 디클(DeCl)	성명	권지아
프로젝트제목	디클		

자체
평가

프로젝트 수행 소감:

기획부터 개발과 배포까지 매주 2회씩 회의를 하며 정성을 쏟아 준비한 프로젝트가 끝나 정말 뿌듯하다.

기획 단계에서는 서비스 컨셉 및 세부 기획에서부터, 우리 서비스가 유사한 다른 서비스(에브리타임, 캠퍼스픽)와 어떻게 차별화될 수 있는지, 그리고 학기 일정을 고려할 때 팀의 개발 목표를 어디까지 두는 것이 좋을지를 신중하게 고려하여 결정하였고, 업무를 나누어 팀원들 각자의 재능과 흥미에 맞는 역할을 분배하고자 노력하였다.

그리고 개발 단계에서 Flutter를 활용한 프론트엔드 개발을 맡아 수행했다. 로그인, 회원가입, 회원 정보 관리(마이페이지) 및 커뮤니티의 게시판 메인 페이지와 카테고리 필터링, 검색 기능을 맡았고, 화면단 구성부터 페이지네이션 등 동적인 요소들까지 모두 개발하였다. 또한 백엔드와 api 스펙을 적절히 협의하여 프론트엔드 코드와 연동하였다.

또한 AI를 사용한 비속어 필터링 API 개발을 맡아 수행했다. 파이썬과 Scikit-learn, 그리고 로지스틱 회귀 모델을 활용하여 한국어 비속어를 감지하는 머신러닝 모델을 학습시키고, 이 모델을 Fastapi와 연결하여 비속어를 작성하려는 사용자에게 경고 알림을 보내는 api를 개발하였다.

마지막으로 배포 단계에서는 배포를 위해 ios와 안드로이드의 네이티브 코드들을 적절히 수정하여 빌드하고, 이를 앱스토어와 구글 플레이스토어에 올려 여러 번의 심사 과정을 거쳐 배포에 성공했다. 이 과정에서 스토어 등록에 필요한 여러 이미지 및 로고 제작, 영상 제작, 문서 제작, 리젝 대응 등 세부 업무가 많았지만 학기 중 시간을 효율적으로 활용하여 마감 기한 내에 성공적으로 배포를 완료할 수 있었다.

이외에도 팀장으로서 중간 발표 및 최종 발표를 위한 PPT 자료와 발표, 포스터 자료, 팀 보고서 등 문서 작업을 수행하고, 팀의 개발 및 배포 일정을 관리하며 우리 팀의 프로젝트가 기한 내에 원활히 수행될 수 있도록 노력했다.

프로젝트의 진행 과정에서 어려운 문제들을 만날 때 상당히 힘들기도 했지만 모든 팀원들과 함께 노력하여 결국에는 전부 해결해낼 수 있었다. 그리고 이 과정에서 함께 끝까지 노력한다면 어떤 문제라도 이겨낼 수 있다는 자신감을 가질 수 있었고, 팀장으로서의 책임감, 한 학기동안 한명도 빠지지 않고 항상 열심히 참여해준 팀원들에게 감사하는 마음을 키울 수 있었다.

개선요구사항:

추후 실사용자들의 의견을 받아 UI/UX를 더욱 편리하게 개선시킬 수도 있을 것 같다.

또한 업데이트가 있을 경우 강제로 업데이트를 진행하게 할 수 있는 기능을 추가하여 업데이트하는 것도 좋을 것 같다.

발전적 제언:

캡스톤 기간이 끝난 이후에도 서비스를 계속 운영하여 실제 사용자들을 받아보고, 서비스를 발전시켜본다면 더욱 좋은 기회가 될 수 있을 것 같다.

4 참고 문헌

번호	종류	제목	출처	발행년도	저자	기타
1	기술문서	Kotlin JDSL	https://kotlin-jdsl.gitbook.io/docs		Line	
2	기술문서	Flutter – Dart API docs	https://api.flutter.dev/index.html		Google	
3	기술문서	Spring	https://docs.spring.io/spring-framework/reference/index.html		VMWare	
4	기술문서	Kotlin	https://kotlinlang.org/docs/home.html		JetBrains	
5	기술문서	Spring Boot	https://docs.spring.io/spring-boot/docs/current/reference/htmlsingle/		VMWare	
6	기술문서	Spring Data JPA	https://docs.spring.io/spring-data/jpa/reference/index.html		VMWare	
7	기술문서	Terraform	https://developer.hashicorp.com/terraform/docs		HashiCorp	
8	기술문서	Kotest	https://kotest.io/docs/framework/framework.html		Kotest Team	

5 부록

5.1 사용자 매뉴얼 - frontend

(1) 가상 시뮬레이터로 앱을 실행하는 방법

- Prerequisite
 - Flutter 3.13.0
 - Dart 3.1.0
 - 안드로이드 스튜디오

- 시뮬레이터 (혹은 에뮬레이터) 실행
안드로이드 스튜디오에서 device manager -> virtual -> create device -> 실행

- 로컬 실행하기
 - 프론트엔드 파일 경로로 진입

```
cd frontend
```
 - 패키지 설치

```
flutter pub get
```
 - 프로젝트 실행

```
flutter run
```

(2) 안드로이드 : apk 설치 방법

파일 경로 frontend -> android -> app > build.gradle

60~73번째 줄 release key 관련 코드 (signingConfigs 와 buildTypes) 주석처리

- APK 생성

```
flutter build apk --release --target-platform=android-arm64
```

(3) IOS : 앱스토어에서 설치 방법

앱스토어 검색창에 학과별 커뮤니티 검색 후 설치

- 앱스토어 URL
<https://apps.apple.com/kr/app/%EB%94%94%ED%81%B4/id6499332881>

5.2 운영자 매뉴얼 - Backend

- Prerequisite
 - Java 17
 - docker compose

- 로컬 MYSQL 설치하기(M1 기준)
 - 백엔드 파일 경로로 진입

```
cd backend
```

- `docker-compose`를 데몬으로 실행

```
docker-compose up -d
```

- `backend/src/main/resources`에 `application-security.yml` 파일 생성 후 아래 내용 작성

```
aws:
```

```
access-key: <YOUR_AWS_ACCESS_KEY>
```

```
secret-key: <YOUR_AWS_SECRET_KEY>
```

```
s3:
```

```
bucket: <YOUR_BUCKET_NAME>
```

```
region: "ap-northeast-2"
```

- `backend`에서 아래 명령어 실행

```
./gradlew bootRun --args='--spring.profiles.active=local'
```

5.3 배포 가이드

- (1) 안드로이드 앱 APK 빌드 방법

```
flutter build apk --release --target-platform=android-arm64
```

- (2) IOS 앱 배포

XCode 와 App Store Connect에 로그인하여, 신규앱 등록을 거쳐야 한다.

5.4 디클 매뉴얼

1.1 의 프로젝트 개요를 참고하여 앱의 기능들과 화면 구성을 알아보시길 바랍니다.

5.5 테스트 케이스

대분류	소분류	기능	테스트 방법	기대 결과	테스트 결과
회원가입	학교 이메일 인증 기능	학교 이메일 인증	회원가입에서 학교 이메일(아이디) 입력란에 이메일을 적고, 오른쪽 전송버튼을 클릭한다.	메일로 인증코드가 전송된다.	성공
회원가입	회원가입 기능	회원가입	회원가입에서 이메일 인증을 포함하여 모든 폼을 채운후 하단 회원가입 버튼을 클릭한다.	회원가입이 완료되었다는 메시지와 함께 로그인 가능해진다.	성공
회원가입	JWT 기반 API 요청	JWT 기반 API 요청	로그인을 한다.	AccessToken, RefreshToken을 저장하여, 자동으로 로그인된다.	성공
회원가입	아이디 및 비밀번호 찾기	아이디 및 비밀번호 찾기	로그인 화면 하단에 아이디 및 비밀번호 찾기를 한다.	아이디 및 비밀번호를 찾을 수 있다.	성공
게시글 CRUD	게시글 페이지네이션 조회	게시글 페이지네이션 조회	메인화면에서 스크롤하여 게시글을 계속 불러온다.	게시글이 빠르게 화면에 나오며, 스크롤하면 게시글을 계속 불러온다.	성공
게시글 CRUD	게시글 작성 및 이미지 업로드	게시글 작성 및 이미지 업로드	메인페이지에서 우측 하단 플러스 버튼을 클릭 -> 게시글 작성 및 하단 이미지 버튼 클릭, 이미지 선택 후 X 버튼을 클릭하여 잘못 선택한 이미지 삭제 가능	게시글에 이미지를 포함하여 업로드된다. 메인페이지에서 작성한 게시글이 보인다.	성공
게시글 CRUD	게시글 삭제	게시글 삭제	해당 게시글 우측 상단 더보기 버튼 클릭 -> 삭제하기, 또는 수정하기 화면에서 우측 하단 삭제하기 버튼 클릭	게시글이 삭제되고, 메인페이지에서 삭제된 게시글이 없어져 보이지 않게 된다.	성공
게시글 CRUD	게시글 수정 및 이미지 수정	게시글 수정 및 이미지 수정	해당 게시글 우측 상단 더보기 버튼 클릭 -> 수정하기, 제목, 내용, 이미지를 바꿀 수 있음	게시글을 수정하여 수정된 내용이 반영된	성공

				글을 볼 수 있게 된다.	
게시글 CRUD	게시글 검색 기능	게시글 검색 기능	메인 페이지에서 우측 하단 플러스 버튼 위에 돋보기 버튼 클릭 -> 검색창에 원하는 글 검색	검색창에 입력된 글과 연관된 글 리스트가 나온다.	성공
게시글 상세	댓글 CRUD	댓글 CRUD	댓글 생성: 해당 게시글의 하단 바를 클릭하면 키보드가 올라옴 -> 내용 작성 후 하단 우측 종이비행기 버튼 클릭 댓글 수정: 댓글 라인 우측에 더보기 버튼 클릭 -> 수정하기 클릭 -> 해당 댓글 색상이 변경됨 -> 입력창에 수정하려는 글로 작성 후 종이비행기 클릭 댓글 삭제: 댓글 라인 우측에 더보기 버튼 클릭 -> 삭제하기 클릭	댓글을 등록, 읽기, 수정, 삭제가 가능하다.	성공
게시글 상세	대댓글 CRUD	대댓글 CRUD	대댓글 생성: 댓글 라인의 메시지 아이콘 클릭 -> 내용 작성 후 종이비행기 클릭 대댓글 수정: 대댓글 라인 우측에 더보기 버튼 클릭 -> 수정하기 클릭 -> 해당 댓글 색상이 변경됨 -> 입력창에 수정하려는 글로 작성 후 종이비행기 클릭 대댓글 삭제: 댓글 라인 우측에 더보기 버튼 클릭 -> 삭제하기 클릭	대댓글을 등록, 읽기, 수정, 삭제가 가능하다.	성공
게시글 상세	게시글 좋아요 기능	게시글 좋아요 기능	게시글 우측 상단 하트 버튼 클릭	하트 클릭 애니메이션이 재생되며 좋아요 수가 올라간다.	성공
게시글 상세	게시글 스크랩 기능	게시글 스크랩 기능	게시글 우측 상단 별 버튼 클릭	색상이 노란색으로 변하고 스크랩 수가 올라간다.	성공
게시글 상세	비속어 필터링	비속어 필터링 기능	위 표를 참고하여 게시글 작성/수정, 댓글 작성/수정, 대댓글 작성/수정하기	비속어가 포함되어 있어 불이익을 받을 수 있다는 안내를 띄움	성공
마이 페이지	내가 작성한 글 조회	내가 작성한 글 조회	메인페이지에서 우측 상단 사람 버튼 클릭 -> 내가 작성한 글 클릭	내가 작성한 글 조회가 된다.	성공
마이 페이지	내가 댓글단 글 조회	내가 댓글단 글 조회	메인페이지에서 우측 상단 사람 버튼 클릭 -> 내가 댓글단 글 클릭	내가 댓글단 글 조회가 된다.	성공

마이 페이지	내가 스크랩 한 글 조회	내가 스크랩 한 글 조회	메인페이지에서 우측 상단 사람 버튼 클릭 -> 내가 스크랩한 글 클릭	내가 스크랩한 글 조회가 된다.	성공
마이 페이지	전공 변경 기능	전공 변경 기능	메인페이지에서 우측 상단 사람 버튼 클릭 -> 전공 변경하기 클릭	전공이 변경된다.	성공
마이 페이지	비밀번호 변경	비밀번호 변경	메인페이지에서 우측 상단 사람 버튼 클릭 -> 회원 탈퇴하기 클릭	비밀번호가 변경이 된다.	성공
마이 페이지	회원 탈퇴	회원 탈퇴	메인페이지에서 우측 상단 사람 버튼 클릭 -> 회원 탈퇴하기 클릭	회원 탈퇴가 된다.	성공
알림기능	알림기능	알림기능	앱 알림 허용	내 게시글에 댓글, 대댓글이 달리거나, 내 댓글에 대댓글이 달리면 푸시 알림이 온다.	성공
신고 기능	신고 기능	신고 기능	신고하려는 게시글, 댓글, 대댓글 더보기에서 신고하기 버튼 클릭 -> 신고 사유 선택	신고가 된다.	성공
차단 기능	차단기능	차단기능	차단하려는 게시글, 댓글, 대댓글 더보기에서 차단하기 버튼 클릭	차단이 되어 해당 유저의 글이나 댓글, 대댓글이 보이지 않는다.	성공

5.6

디클에 대한 기술 문서

4. 의 참고문헌에 포함되어 있다.

4.2 결과보고서 발표자료

다학제간 캡스톤 디자인 최종발표

7조 Decl팀

20190155 권지아
20171659 윤용배
20212674 김동윤
20213032 윤홍현

목차

CONTENTS

01 프로젝트 기획

프로젝트 Decl의 기획 과정

02 서비스 구조 소개

Decl 사용 기술 및 인프라 구조

03 주요 기능 소개

Decl의 주요 기능

04 배포 현황

애플 앱스토어 및
구글 플레이스토어 배포 현황

프로젝트 기획

01 프로젝트 기획

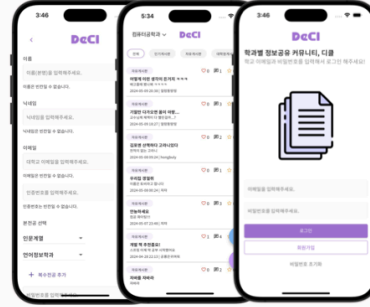
프로젝트 소개

학과별 정보 공유 커뮤니티, 디클

디클은 전국의 대학생들이 학과별로 모여 정보를 공유할 수 있는 커뮤니티 서비스입니다.

전공을 공통 관심사로 가진 대학생들이 모여 좀 더 깊은 관심사를 공유할 수 있는 소통의 장구가 되고자 합니다.

Frontend: Flutter 3.13.0, Dart 3.1.0
Backend: Spring Boot 3.2.3, Kotlin 1.9.0, Kotlin-jdsl 3.3.1
Infrastructure: Terraform, AWS ECS + Fargate
CI/CD: github action



시장 포지셔닝

01 프로젝트 기획




x: 유저들이 모이는 기준
y: 같은 공통 관심사를 가진 유저 간의 소통


사용 기술

02 서비스 구조 소개


AI




Frontend



Backend



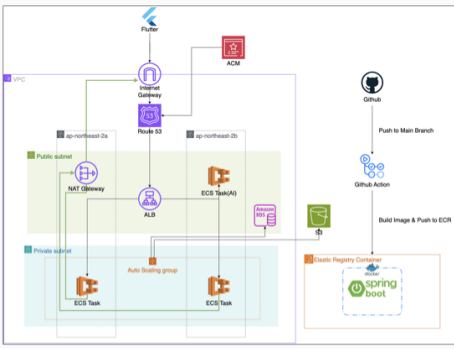
Infrastructure



Frontend: Flutter 3.13.0, Dart 3.1.0
Backend : Spring Boot 3.2.3, Kotlin 1.9.0, Kotlin-jdsl 3.3.1
Infrastructure : Terraform, AWS ECS + Fargate
AI: Python 3.12.3, Scikit learn
CI/CD : github action

서비스 구조도

02 서비스 구조 소개



A. 가용성
두개의 가용영역을 사용해 가용성을 높임

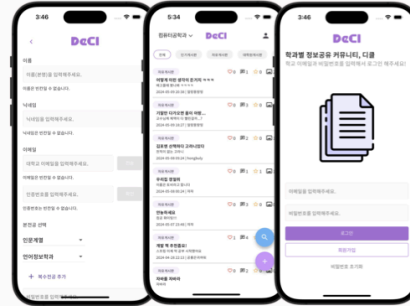
B. 확장성
로드밸런서와 오토 스케일링을 활용하여 스프링 서버의 메모리,CPU 사용량이 많아지면 컨테이너를 추가로 띄워 scalability를 개선함.

디클 주요 기능

03 주요 기능 소개

1. 회원가입 및 로그인

- 전국 383개 대학교의 학생들이 자신의 과와 가장 유사한 학과를 직접 선택(최대 2개), 학교 이메일로 인증 후 가입
- 학교 이메일을 통해 로그인 가능

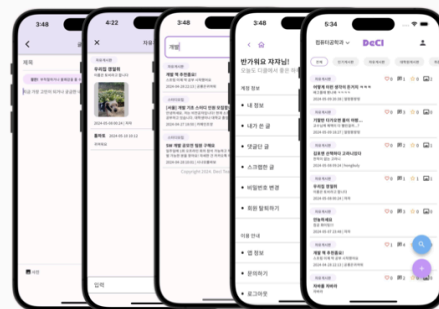


디클 주요 기능

03 주요 기능 소개

2. 게시글, 댓글 CRUD 및 검색, 마이페이지 기능

- 사용자가 자유롭게 글과 댓글을 작성, 수정, 삭제, 조회할 수 있음
- 다양한 카테고리 필터링 기능 제공
- 게시글 검색 기능 제공
- 사진 확대 및 넘기기 기능 제공
- 좋아요, 댓글, 대댓글, 스크랩 기능 구현
- 마이페이지에서 회원정보 조회, 전공 변경, 비밀번호 초기화/변경, 회원 탈퇴, 이메일 문의하기 등 다양한 회원 지원 기능을 제공



디클 주요 기능

03 주요 기능 소개

3. 부적절한 콘텐츠 신고 및 유저 차단 기능

- 불쾌한 콘텐츠를 신고할 수 있도록 신고 기능 구현
- 신고가 10회 이상 누적될 경우, 콘텐츠가 삭제 처리되고 해당 콘텐츠의 작성자는 블랙리스트에 올라 90일간 게시물, 댓글 작성이 불가함.
- 불쾌한 콘텐츠를 생성하는 유저를 차단할 수 있는 기능 구현



디클 주요 기능

03 주요 기능 소개

4. AI 활용 비속어 필터링 기능

- 파이썬 Scikit-learn 라이브러리와 로지스틱 회귀 모델을 활용하여 한국어 비속어 여부를 판별하는 모델 학습 (accuracy: 0.89)
- fastapi를 사용하여 요청에 담긴 메시지 값이 비속어인지 아닌지를 응답으로 주는 api 생성
- 사용자가 글과 댓글에 비속어로 추정되는 텍스트를 작성할 경우 해당 부분에 대해 경고 알림을 보내도록 구현



디클 주요 기능

03 주요 기능 소개

5. SSE를 활용한 실시간 댓글 알림 기능

- SSE(Server-Sent-Event) 단방향 통신을 이용하여 댓글 알림을 클라이언트에게 실시간으로 전송
- 지속적으로 하트비트를 보내 클라이언트와 연결이 끊어질 경우 다시 연결을 맺을 수 있도록 구현
- 알림을 클릭할 경우 해당 게시글의 상세 페이지로 가도록 구현



디클 주요 기능

03 주요 기능 소개

목표 개발 기능 (목표 대비 100% 완료)

- 이메일 인증, 회원가입 및 로그인
- 게시글 CRUD 및 검색 기능
- 마이페이지 기능
- 회원 정보 변경 및 지원 기능
- SSE 실시간 댓글 알림 기능

추가 개발 기능

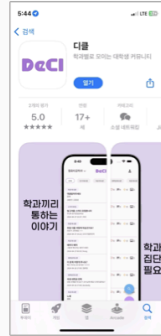
- AI 활용 비속어 감지 기능
- 부적절한 게시글 신고 기능
- 특정 유저 차단 기능

배포 현황

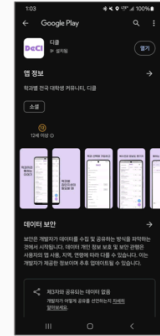
04 배포 현황

앱 배포

- 애플 앱스토어 배포 완료



- 구글 플레이 스토어 배포 완료



역할 분담

권지아(20190155)

- 팀장 (중간발표, 최종발표 PPT, 포스터 제작 및 발표)
- 프론트엔드 개발 (로그인/회원가입/비밀번호 초기화 및 변경/ 마이페이지 개발-내가쓴 글, 댓글단 글, 스크랩한 글, 전공 수정, 문의하기, 탈퇴하기 등/게시판 메인 페이지 및 카테고리 필터링, 활성화된 전공 변경기능 개발/검색 기능 개발)
- AI 비속어 필터링 기능 개발 (AI 모델 학습 및 API 개발)
- 앱스토어, 플레이스토어 배포 담당

김동윤(20212674)

- 백엔드 개발 (로그인 / 비밀번호 초기화 / 유저 정보 조회 및 수정 / 게시글 / 검색 및 인기글, 댓글 & 대댓글 단 글, 스크랩한 글 조회 / 댓글&대댓글 / 알림 / 유저 정치 / 회원 탈퇴기능 개발)

윤종배(20171659)

- 백엔드 개발 (신고, 차단, 게시글, 좋아요, 스크랩 기능 개발)
- 백엔드 도메인 설계
- 대학교 이메일 정보 수집 및 이메일 발송 기능 개발
- 이미지 업로드 기능 개발
- CI, CD 구축 및 AWS 서버 배포 담당

윤홍현(20213032)

- 프론트엔드 개발 (상세페이지 / 댓글, 대댓글 기능 / 좋아요 스크랩 기능 / 신고 및 차단기능 / 사진확대 및 슬라이드 기능 / 글작성 및 수정페이지 / 사진 업로드 기능 / SSE 댓글 알림기능 개발)
- UI/UX 담당

7조 발표를 마칩니다.
들어주셔서 감사합니다.

발표자 권지아

7조 Daei

제 5 장

회의록

2024.03.22

캡스톤 회의록 7조

권지아, 김동윤, 윤웅배, 윤홍현 (작성자: 권지아)

교수님과의 면담

중간 발표때 포함하면 좋을 내용을 설명해주셨습니다.

서비스의 설명과 사진, 우리 서비스만의 차별점, 우리 서비스가 어떤 의미를 가지는지,

다이어그램, 아키텍처 설명 그림 등을 포함하면 좋을 것이라고 말씀해주셔서 적극 반영하고자 합니다.

ai를 활용한 기능으로 학과별 관심 콘텐츠 추천, 비속어 필터링 등을 고려하며 함께 논의하고 있습니다.

회의 내용

- AI 도입 여부, 프로젝트 방향성 논의
- 게시판 기능 (메인 기능) 관련 api 스펙 논의
- 프론트엔드와 백엔드 남은 기능 논의
 - 프론트엔드
 - 글 작성 폼 ui + api 연결
 - 글 수정 api 연결
 - 글 상세 폼 ui + api 연결
 - 글 삭제 api 연결
 - 리스트에서 글 상세 페이지 들어가게 하기
 - 마이페이지 내가 쓴 글
 - 마이페이지 댓글 단 글
 - 마이페이지 스크랩 글 페이지
 - 위로 당기면 refetching되게 구현
 - 리스트에서 학과 변경하고 리스트 새로 불러올 수 있게 구현
 - 기본 학과 2개 선택 → 변경기능
 - 아이디 찾기 기능
 - 비밀번호 초기화 기능
 - 백엔드
 - 커스텀 예외 로직 작성하기
 - 다음 주 테라폼 배포 논의
 - 게시글이미지 수정 로직 작성하기
 - 테스트 커버리지 높이기
 - CD 방법 공부하기

캡스톤 디자인 7조 4주차 팀 회의록

회의록을 작성한 사람

- 윤웅배

회의 일시

- 2024.03.29

회의 참석자

- 권지아, 윤웅배, 윤홍현, 김동윤

회의 내용

- 게시판 작성 페이지 UI
- 리팩토링 관점에서 커스텀 예외 사용 vs 기본 예외 사용 논의(커스텀 예외로 결정)
- CI / CD 구축
- 활성화된 학과 오류 수정 논의

교수님과의 면담

- 기본 기능 완성 후 추가 기능을 고려했으면 좋겠다.

회의록 작성자의 의견과 소감

- 기본 기능이 거의 끝나 다음 회의 때 추가기능을 논의 해야겠다는 생각이 들었습니다.
- 매주 회의를 통해 서로가 개발 했던 것에 대해 팔로우업 되어 좋았습니다.

캡스톤 디자인 7조 6주차 팀 회의록

회의록을 작성한 사람

- 윤웅배

회의 일시

- 2024.04.12

회의 참석자

- 권지아, 윤웅배, 윤홍현

회의 내용

- 비속어 필터링을 백엔드에서 아호코라식을 통해 구현할지, 파이썬 서버에서 머신러닝 모델을 활용해서 구현할지 논의 → 파이썬을 통해 구현하기로 결정됨
- SSE 알림 프론트에서 오류 발생, 원인 파악 중
- 페이지네이션 성능 논의

교수님과의 면담

- 중간발표때 세션 교수님들께 받았던 피드백을 설명드렸습니다. 현재 진행상황에 대해 설명드리고 AI를 활용해 서비스를 개선하면 좋겠다는 피드백을 받았습니다. 그래서 현재 머신러닝 모델을 활용한 비속어 필터링 기능을 개발중에 있다고 설명드렸습니다.

회의록 작성자의 의견과 소감

- 팀장님이 회의 진행과 의견 조율을 잘 해주셔서 토의가 잘 이뤄짐
- 프로젝트가 수월하게 진행됨에 따라 다음 일정도 무리 없이 소화 가능할 것으로 예상된다.

캡스톤 디자인 7조 7주차 팀 회의록

회의록을 작성한 사람

- 윤홍현

회의 일시

- 2024.04.19

회의 참석자

- 권지아, 윤웅배, 윤홍현, 김동윤

회의 내용

- 버그 내용 및 보완할 기능 회의
- 비속어 필터링(로지스틱회귀모델) 기능이 비속어가 아닌 문장도 비속어로 분류하는 문제가 발생해 사용여부 의논

버그 목록

- 내가 쓴 글, 댓글단 글, 스크랩한 글 업데이트가 바로 안됨
 - 글, 댓글 작성, 스크랩 후 세가지 다 들어가보면, 조회가 바로 되지 않음 (시간 좀 걸림)
 - 삭제 및 스크랩 취소 시에도, 없어지지 않고 위 세 군데엔 여전히 남아있음
- ~~학과 변경 후, 다시 원래 학과로 변경할 시 (컴공 → 경영 → 컴공), 게시글 몇개가 조회가 안됨 (일부만 조회 됨) → 새로고침 해야만 다시 게시글들 조회 됨.~~

- 게시글 작성 후, 작성된 게시글이 목록에 바로 조회되지 않는 경우가 있음 (가끔)
- 게시글의 댓글 개수 카운트 하는 부분에 대댓글 개수가 포함이 안됨
- 자기 댓글에 좋아요 누르면 오류 메시지 뜰때 오버플로우
- 사진 오류 (해시함수로 값 생성해서 보내기)
- 댓글 달릴때 순서가 이상하게 생기는 문제

보완할 기능

- 맨 위로 스크롤시 및 **Decl** 로고 클릭시 게시판 새로고침
- 이미지 여러개 조회시 스와이프로 다음 이미지 넘어갈 수 있도록
- 댓글에 작성 시간 추가하기
- 시간 포맷 변경
- **input** 바 패딩 일정하게 통일
 - 로그인, 회원가입, 비밀번호 초기화 시 입력할때 글자가 너무 왼쪽에 붙어있음. 패딩을 좀 넣으면 좋을듯
- 회원가입 시에 입력되지 않은 칸이 있으면, 입력하지 않은 칸으로 자동으로 스크롤 **or** 다른색으로 표시 **or** 경고창을 띄워주는 액션이 있으면 좋을것 같음
- 좋아요 및 스크랩이 너무 작은것 같음
- 게시글 수정 시에는 카테고리 설정하는 부분이 안보여야 할듯, 그리고 상단에 “글 작성” 이 아닌 “글 수정” 이런 식으로 들어가는것이 좋을듯.
- 글자랑 아이콘이 다 너무 검은색이라 에타 처럼 다른 색깔을 섞는것이 좋아보임
- 비밀번호 변경, 검색, 알림 기능

교수님과의 면담

- 일주일 단위로 가장 어필할 수 있는 기능 구현
- 비속어 **AI** 필터링 기능을 버리지 말고, 새로운 모델이나 학습을 더 시켜서 발전을 하길 권장
- 새로운 차별점을 줄 수 있는 기능이나 **UI**를 개선
- 기본적으로 에러나 디버깅은 계속하고 **AI**기능은 한명이 계속 해야함

캡스톤 디자인 7조 6주차 팀 회의록

회의록을 작성한 사람

- 김동윤

회의 일시

- 2024.05.03

회의 참석자

- 권지아, 윤웅배, 윤홍현, 김동윤

회의 내용

- 버그 및 보완할 기능 조치 결과 보고
- 앱 스토어 & 플레이 스토어 배포 논의
 - 앱 스토어 배포 신청 완료 -> 반려됨
 - 플레이 스토어 배포 진행중
- 앱 스토어 배포 신청 반려 원인 해결 논의
 - 부적절한 유저 및 콘텐츠 필터링 기능 개발 필요
 - 신고 기능을 구현하여 해결 예정

버그 해결 목록

- 내가 쓴 글, 댓글단 글, 스크랩한 글 업데이트가 바로 안됨 - 진행완료(지아)
 - 글, 댓글 작성, 스크랩 후 세가지 다 들어가보면, 조회가 바로 되지 않음 (시간 좀 걸림)
 - 삭제 및 스크랩 취소 시에도, 없어지지 않고 위 세 군데엔 여전히 남아있음
- 게시글 작성 후, 작성된 게시글이 목록에 바로 조회되지 않는 경우가 있음 (가끔) → 업로드 후 등록됐다는 응답을 받고, 게시글 목록을 refresh하는데 조회가 안되는 거라 서버에서 게시글 목록을 빠르게 추가해서 넘겨주질 못하는 것이 아닐까... → 확인해보니 이제 문제없는듯 합니다!
- 학과 변경 후, 다시 원래 학과로 변경할 시 (컴공 → 경영 → 컴공), 게시글 몇개가 조회가 안됨 (일부만 조회 됨) → 새로고침 해야만 다시 게시글들 조회 됨. - 진행완료(지아)
- 게시글의 댓글 개수 카운트 하는 부분에 대댓글 개수가 포함이 안됨 - 진행완료
- 자기 댓글에 좋아요 누르면 오류 메시지 뜰때 오버플로우 - 진행완료(윤홍현)
- 사진 오류 (해시함수로 값 생성해서 보내기) - 진행완료(윤홍현)
- 댓글 달릴때 순서가 이상하게 생기는 문제 - 백엔드 문제로 확인됨, 진행완료
- 댓글에 내가 하트 눌렀는지 isLiked 값 제대로 안된 것들이 있음
- 키보드 올라왔을 때 화면 overflow - 진행완료(윤홍현)
- 댓글 공백이어도 작성됨 - 진행완료(윤홍현)
- 욕설이어도 등록할 수 있게 - 진행완료(윤홍현)
- 게시판 변경 및 질문글 등록이 안됨 - 완료 (질문글 삭제)
- 질문게시판이 안나옴 - 완료 (질문글 삭제)
- 질문글로 변경 불가능 - 완료 (질문글 삭제)
- 토론 재발급이 안되는 문제 (401) - 진행완료 (지아)
- 페이지네이션 내린 상태에서 글 작성 완료하면 리스트 제대로 refresh 안되는 문제 - 진행완료(지아)
- 비속어 필터링의 팝업의 오버플로우 문제 수정 - 진행완료(지아)

보완한 기능 목록

- 맨 위로 스크롤시 및 Decl 로고 클릭시 게시판 새로고침 - 진행완료(지아)
- 이미지 여러개 조회시 스와이프로 다음 이미지 넘어갈 수 있도록 - 진행완료(윤홍현)
- 댓글에 작성 시간 추가하기 - 진행완료(윤홍현)
- 시간 포맷 변경 - 진행완료(윤홍현)
- input 바 패딩 일정하게 통일
 - 로그인, 회원가입, 비밀번호 초기화 시 입력할때 글자가 너무 왼쪽에 붙어있음. 패딩을 좀 넣으면 좋을듯 - 진행완료(지아)
- 회원가입 시에 입력되지 않은 칸이 있으면, 입력하지 않은 칸으로 자동으로 스크롤 or 다른색으로 표시 or 경고창을 띄워주는 액션이 있으면 좋을것 같음 - 진행완료(지아)
- 좋아요 및 스크랩이 너무 작은것 같음 - 진행완료(윤홍현)
- 상단에 “글 작성” 이 아닌 “글 수정” 이런 식으로 들어가는것이 좋을듯. - 진행완료(윤홍현)
- 글자랑 아이콘이 다 너무 검은색이라 에타 처럼 다른 색깔을 섞는것이 좋아보임 - 진행완료(지아)
- 비밀번호 변경 기능 (/edit-password 경로 oldPassword, password, confirmPassword - POST + accessToken) - 진행완료(지아)

- 게시글 검색 기능 - 진행완료 (지아)
알림 기능 - 진행완료 (동윤+홍현)
메인페이지 상단바 양옆 간격이 왼쪽보다 오른쪽이 더 넓어보임 - 진행완료(지아)
- 로그인 회원가입 등 버튼 디자인 고정 - 진행완료(지아)
- 검색 버튼 디자인 고정 - 진행완료(지아)
- **floating button** 디자인 - 진행완료 (윤홍현)

교수님과의 면담

- 최종 발표시 어떻게 구현한 기능들을 잘 어필할 수 있을지 논의
- 서비스의 안정성에 관련하여 보완

회의록 작성자의 의견과 소감

- 앱 스토어 배포 신청 반려 원인을 해결하면 금방 배포가 될 것으로 보임.
- 배포 완료 후, QA 및 발표 자료 준비를 시작 하면 될 것 같다고 생각됨

캡스톤 디자인 7조 7주차 팀 회의록

회의록을 작성한 사람

- 윤홍현

회의 일시

- 2024.05.10

회의 참석자

- 권지아, 윤웅배, 윤홍현, 김동윤

회의 내용

- 안드로이드 리젝 사유 확인

교수님과의 면담

- 배포를 했다는 점을 강조하기 위해 앱스토어에서 다운받아서 진행하는 모습을 촬영
- 서버와 실제 단말기에서 통신되는 부분, 단말기에서 요청하면 서버에서 로그가 뜨는 것을 같이 촬영
- 홍보를 많이 하여 실사용자 확보 필요

캡스톤 디자인 7조 11주차 팀 회의록

회의록을 작성한 사람

- 김동윤

회의 일시

- 2024.05.017

회의 참석자

- 권지아, 윤웅배, 윤홍현, 김동윤

회의 내용

- 플레이스토어 앱 배포 완료

- 포스터 작업
- 발표 자료 작업
- 최종 및 수행 보고서 작성
- 앱 홍보 방법 논의

교수님과의 면담

- 피피티 색상 회색으로 변경
- 중요한 기능들 위주로 설명하고 회원기능 축소

제 6 장

프로젝트 수행

자체 평가

자 체 평 가 서

팀명	7조	성명	윤홍현
프로젝트제목	디클 (Decl)		
자체 평가	<p>프로젝트 수행 소감:</p> <p>하이브리드 앱 개발을 위해 Flutter를 배운 것은 새로운 기술을 배우는 것에 대한 두려움을 줄이는 데 도움이 되었습니다. 앱 개발에서 게시글 상세페이지, 게시글 등록페이지를 담당하여 개발하였습니다. 게시글 상세페이지에서는 댓글/대댓글 페이지네이션과 댓글, 대댓글 달기, 수정, 삭제를 구현하고, 신고하기/차단하기 등 백엔드와 연결하여 필요한 기능들을 구현하였습니다.</p> <p>평소에 팀프로젝트를 하면, 적극적이지 않고 자신이 해야 할 일들을 다 하지 못하는 팀원들과 프로젝트를 진행하여, 거의 대부분을 제가 프로젝트를 했기에, 팀으로 일하는 것을 싫어했습니다. 하지만 이번 캡스톤에서는 저에게 필요한 점들을 짚어주고, 도움을 받을 수 있는 팀원이 함께 하여, 기술을 빠르게 습득하고 개발을 하는데 도움이 되었습니다. 이러한 경험을 통해 같이 일하는 팀원의 책임감이나 능력이 중요하다는 점을 다시한번 느꼈습니다. 저에게 이 점을 적용하여 함께 일하고 싶고, 함께 성장하고 싶은 팀원이 될 수 있도록 노력해야 겠다고 느꼈습니다.</p> <p>개선요구사항:</p> <p>프로젝트를 하면서, 저 자신에게 개선해야겠다고 느낀 점은 깃 사용이었습니다. 오류를 만났을 때 수정을 해야 하는데, 깃 커밋을 쪼개지 않고 계속 작업을 하는 버릇이 있었습니다. 앞으로 이러한 습관을 개선해야 겠다고 느꼈습니다.</p> <p>프로젝트에서 개선할 점으로는 코드를 간략하게 하고, 로딩시간을 줄이기 위한 방법을 더 모색해야 겠다고 느꼈습니다.</p> <p>발전적 제언:</p> <p>하이브리드 앱 개발 말고 네이티브 앱 개발에서만 할 수 있는 것들이 있기 때문에, 네이티브 앱 개발 중 IOS에 대해 더 공부해 봐야겠다고 느꼈습니다.</p> <p>프로젝트에서 발전해야 할 점으로는 대학교 어플이니까 시간표와 식사표가 같이 있는것이 사용자들이 더 앱을 자주 사용하게 만들 수 있는 기능이기에 때문에 추가적으로 있으면 좋겠다고 생각했습니다.</p>		

자 체 평 가 서

팀명	7조	성명	윤웅배
프로젝트제목	디클		
자체 평가	<p>프로젝트 수행 소감:</p> <p>이전 프로젝트를 수행하며 개선하거나 새로 적용해보고 싶은 기술들이 여럿 있었습니다. 대표적으로 Kotlin + Spring, CI/CD, 도메인 주도 개발, 테스트 커버리지, Terraform 활용뿐만 아니라 브랜치 전략과 깃허브를 활용한 협업을 경험해보고 싶었습니다. 캡스톤 프로젝트에서 그 기회가 닿아 새로운 기술들을 적용해볼 수 있었습니다. 피상적으로 알고만 있던 지식들을 실제 코드로 구현하는 과정에서 미처 알지 못했던 여러 문제들을 겪었고 이를 해결하는 과정에서 성장할 수 있어 좋았습니다. 특히 테라폼을 활용해 AWS에 배포하는 경험을 통해 AWS에서 제공하는 서비스에 대해 많은 공부를 할 수 있어 좋았습니다. 개발부터 앱스토어에 배포하는 전체 프로세스를 체험하고 나아가 이를 유지보수하며 실제 현업에 가까운 경험을 할 수 있는 소중한 기회라고 생각합니다. 개발 외적으로도 좋은 팀원들과 의견 공유와 토의하는 과정에서도 개인적 성장을 이루었다 생각합니다.</p> <p>개선요구사항:</p> <p>모든 부분에서 클린 코드 원칙을 지키며 코드를 작성하고 싶었지만 더 나은 코드 작성 방법을 알지 못해 어쩔 수 없이 원칙을 어겼던 적이 있습니다. 기회가 된다면 이런 부분도 리팩토링 하고 싶습니다.</p> <p>발전적 제언:</p> <p>협업 프로세스를 개선해 매주 회의를 갖는 대신 매일 짧은 회의를 가져 모든 팀원이 동일한 정보를 팔로우업 할 수 있도록 한다.</p>		

자 체 평 가 서

팀명	Decl	성명	김동윤
프로젝트제목	Decl		
자체 평가	<p>프로젝트 수행 소감:</p> <p>이번 프로젝트는 제가 백엔드 개발자로 참여하는 두 번째 프로젝트였습니다. 백엔드 개발에 대한 이해도가 여전히 부족하다고 느끼는 가운데, 이번 프로젝트를 통해 처음으로 다루어 보는 프레임워크와 프로그래밍 언어에 도전했습니다. 프로젝트 시작 전, 저는 새로운 기술 습득에 많은 시간을 할애했습니다. 이 과정에서 기본 개념을 익히는 것뿐만 아니라, 다양한 개발 도구와 라이브러리에 대한 이해도를 높이기 위해 노력했습니다.</p> <p>프로젝트를 진행하면서, 같은 백엔드 포지션에 있는 팀원으로부터 많은 것을 배울 수 있었습니다. 그들의 경험과 지식 덕분에 개발 과정 중 발생할 수 있는 다양한 문제들을 쉽게 해결할 수 있었습니다. 특히, 새로운 프레임워크와 언어를 사용함에도 불구하고, 프로젝트를 성공적으로 마무리할 수 있었던 것은 큰 성취감을 안겨주었습니다. 이 경험은 저에게 많은 자신감을 주었고, 앞으로의 개발자 생활에 긍정적인 영향을 미칠 것입니다.</p> <p>그러나, 프로젝트의 개발 부분 외에 인프라스트럭처 설계와 프로젝트 세팅에 직접 참여하지 못한 것은 아쉬움으로 남습니다. 이 부분은 프로젝트의 성공적인 진행을 위해 매우 중요한 요소임에도 불구하고, 저는 이번 기회에 그에 대한 경험을 충분히 쌓지 못했습니다. 하지만, 이러한 아쉬움도 팀원으로부터 관련 지식을 배울 수 있는 기회가 되었습니다. 이 경험을 바탕으로, 다음 프로젝트에서는 프로젝트 세팅과 인프라스트럭처 설계를 주도적으로 진행할 수 있는 능력을 갖추고자 합니다.</p> <p>또한, 이번 프로젝트를 통해 처음으로 앱스토어와 플레이스토어에 앱을 출시하는 경험을 할 수 있었습니다. 실제 사용자들이 저희 앱을 다운로드하고 사용하는 것을 보며, 개발자로서 큰 뿌듯함을 느꼈습니다. 이번 경험은 저에게 실제 사용자를 대상으로 한 애플리케이션 개발의 중요성과 그에 따른 책임감을 일깨워 주었습니다.</p> <p>앞으로 저는 더 많은 사용자를 확보하고, 높은 트래픽을 관리하며, 애플리케이션의 지속적인 유지보수를 진행하는 경험도 쌓고 싶습니다. 이번 프로젝트가 저에게 많은 것을 가르쳐 준 만큼, 다음 프로젝트에서는 이러한 경험을 바탕으로 더욱 성장한 모습을 보여주고 싶습니다. 프로젝트를 통해 배운 교훈과 경험은 제 개발자 생활의 소중한 자산이 될 것이며, 앞으로 더 좋은 개발자로 성장하는 데 큰 도움이 될 것입니다.</p> <p>개선요구사항:</p> <p>4. 인프라스트럭처 설계 지식 강화 및 실습: 프로젝트 초기 단계에서 인프라스트럭처 설계에 더 적극적으로 참여하고 싶습니다. 이를 위해 클라우드 서비스(AWS, GCP 등)에</p>		

대한 심화 학습을 진행하며, 인프라스트럭처를 구성하는 다양한 서비스들(예: 컴퓨팅, 스토리지, 네트워킹 등)의 조합과 최적화 방법을 습득할 계획입니다. 또한, 실제 프로젝트에서 인프라 설계를 담당하여, 이론과 실습을 병행하는 경험을 쌓고자 합니다.

5. **대용량 데이터 처리 최적화:** 대규모 트래픽과 대용량 데이터 처리는 현대 애플리케이션 개발에서 중요한 요소입니다. 이에 대응하기 위해, 데이터베이스 쿼리 최적화와 도메인 구조 개선에 주력할 예정입니다. **SQL** 쿼리 튜닝, 인덱싱 전략 수립, **NoSQL** 데이터베이스의 적절한 활용 등을 통해 데이터 조회 성능을 개선하고자 합니다. 또한, 캐싱, 데이터 파티셔닝, 비동기 처리 등의 기법을 도입하여 시스템의 전반적인 성능을 향상시킬 계획입니다.
6. **코드 품질 개선 및 리팩토링:** 코드의 중복을 최소화하고, 가독성과 유지보수성을 높이는 것은 지속 가능한 개발을 위해 필수적입니다. 이를 위해, 리팩토링에 좀 더 시간을 할애하며, 디자인 패턴, 함수형 프로그래밍 등의 원칙을 적용하여 더 깔끔하고 효율적인 코드를 작성할 계획입니다. 코드 리뷰 문화를 활성화하여 팀 내에서의 코드 품질 향상을 도모하고, 지속적인 학습을 통해 새로운 기술과 방법론을 접하며 개발 역량을 강화할 예정입니다.

발전적 제언:

4. **테스트 코드 작성 및 테스트 커버리지 향상:** 프로젝트의 안정성을 확보하기 위해 모든 기능에 대한 충분한 테스트 코드를 작성하는 것이 중요합니다. 이를 통해 코드 변경 시 발생할 수 있는 예기치 않은 오류를 사전에 방지하고, 높은 테스트 커버리지를 유지하여 신뢰성 있는 서비스를 제공할 수 있습니다. 테스트 주도 개발(**TDD**) 방식을 도입하여 개발 과정에서부터 테스트를 중심으로 코드를 작성하는 방법도 고려해볼 수 있습니다.
5. **주기적인 코드 리뷰 실시:** 코드 리뷰는 코드의 품질을 향상시키고, 개발자간의 지식 공유에도 매우 효과적인 방법입니다. 주기적으로 코드 리뷰를 실시하여 서로의 코드에 대한 구체적이고 건설적인 피드백을 제공함으로써, 개선된 코드 작성을 도모하고 개발자의 역량을 함께 향상시킬 수 있습니다. 특히, 신규 기능 개발이나 큰 변경 사항이 있는 경우에는 반드시 코드 리뷰를 통해 검증하는 과정을 거치는 것이 좋습니다.
6. **정기적인 진행 상황 공유 및 계획 토론:** 프로젝트의 진행 상황을 정기적으로 공유하고, 팀원 간의 의견을 나누며, 향후 계획에 대해 논의하는 것은 프로젝트를 계획대로 성공적으로 마치는 데 필수적입니다. 이러한 정기적인 회의를 통해 프로젝트의 목표 달성 상황을 점검하고, 문제가 발생했을 때 신속하게 대처할 수 있는 방안을 모색하며, 미래에 발생할 수 있는 잠재적 문제에 대해 사전에 대비할 수 있습니다.

자 체 평 가 서

팀명	7조 디클(DeCl)	성명	권지아
프로젝트제목	디클		

자체
평가

프로젝트 수행 소감:

기획부터 개발과 배포까지 매주 2회씩 회의를 하며 정성을 쏟아 준비한 프로젝트가 끝나 정말 뿌듯하다.

기획 단계에서는 서비스 컨셉 및 세부 기획에서부터, 우리 서비스가 유사한 다른 서비스(에브리타임, 캠퍼스픽)와 어떻게 차별화될 수 있는지, 그리고 학기 일정을 고려할 때 팀의 개발 목표를 어디까지 두는 것이 좋을지를 신중하게 고려하여 결정하였고, 업무를 나누어 팀원들 각자의 재능과 흥미에 맞는 역할을 분배하고자 노력하였다.

그리고 개발 단계에서 Flutter를 활용한 프론트엔드 개발을 맡아 수행했다. 로그인, 회원가입, 회원 정보 관리(마이페이지) 및 커뮤니티의 게시판 메인 페이지와 카테고리 필터링, 검색 기능을 맡았고, 화면단 구성부터 페이지네이션 등 동적인 요소들까지 모두 개발하였다. 또한 백엔드와 api 스펙을 적절히 협의하여 프론트엔드 코드와 연동하였다.

또한 AI를 사용한 비속어 필터링 API 개발을 맡아 수행했다. 파이썬과 Scikit-learn, 그리고 로지스틱 회귀 모델을 활용하여 한국어 비속어를 감지하는 머신러닝 모델을 학습시키고, 이 모델을 Fastapi와 연결하여 비속어를 작성하려는 사용자에게 경고 알림을 보내는 api를 개발하였다.

마지막으로 배포 단계에서는 배포를 위해 ios와 안드로이드의 네이티브 코드들을 적절히 수정하여 빌드하고, 이를 앱스토어와 구글 플레이스토어에 올려 여러 번의 심사 과정을 거쳐 배포에 성공했다. 이 과정에서 스토어 등록에 필요한 여러 이미지 및 로고 제작, 영상 제작, 문서 제작, 리젝 대응 등 세부 업무가 많았지만 학기 중 시간을 효율적으로 활용하여 마감 기한 내에 성공적으로 배포를 완료할 수 있었다.

이외에도 팀장으로서 중간 발표 및 최종 발표를 위한 PPT 자료와 발표, 포스터 자료, 팀 보고서 등 문서 작업을 수행하고, 팀의 개발 및 배포 일정을 관리하며 우리 팀의 프로젝트가 기한 내에 원활히 수행될 수 있도록 노력했다.

프로젝트의 진행 과정에서 어려운 문제들을 만날 때 상당히 힘들기도 했지만 모든 팀원들과 함께 노력하여 결국에는 전부 해결해낼 수 있었다. 그리고 이 과정에서 함께 끝까지 노력한다면 어떤 문제라도 이겨낼 수 있다는 자신감을 가질 수 있었고, 팀장으로서의 책임감, 한 학기동안 한명도 빠지지 않고 항상 열심히 참여해준 팀원들에게 감사하는 마음을 키울 수 있었다.

개선요구사항:

추후 실사용자들의 의견을 받아 UI/UX를 더욱 편리하게 개선시킬 수도 있을 것 같다.

또한 업데이트가 있을 경우 강제로 업데이트를 진행하게 할 수 있는 기능을 추가하여 업데이트하는 것도 좋을 것 같다.

발전적 제언:

캡스톤 기간이 끝난 이후에도 서비스를 계속 운영하여 실제 사용자들을 받아보고, 서비스를 발전시켜본다면 더욱 좋은 기회가 될 수 있을 것 같다.

주 의 문

2024년 캡스톤 디자인 I 종합설계 프로젝트 최종보고서
디클

발행인 : 권지아, 윤홍현, 윤웅배, 김동윤

발행일 : 2024년 6월 14일

발행처 : 국민대학교 소프트웨어학부 소프트웨어학과

주소 : (136-702) 서울시 성북구 정릉동 861-1

- 이 보고서는 국민대학교 전자정보통신대학 컴퓨터공학부에서 개설한 교과목 캡스톤 디자인 I 에서 수행한 프로젝트 최종보고서입니다.
 - 이 보고서의 내용은 국민대학교 컴퓨터공학부 및 위 발행인들의 서면 허락없이 사용되거나, 재가공 될 수 없습니다.
-