



국민대학교
소프트웨어융합대학
소프트웨어학부

캡스톤 디자인 I

종합설계 프로젝트

프로젝트 명	지하철 커뮤니티 - 출그니
팀 명	28조 - 출그니
문서 제목	결과보고서

Version	1.2
Date	2023-05-22

팀원	신 명철 (조장)
	송 민준
	서 민석
지도교수	김 장호 교수

 국민대학교 소프트웨어학부 캡스톤 디자인 I	수행보고서		
	프로젝트 명	지하철 커뮤니티 - 출그니	
	팀 명	28조	
	Confidential Restricted	Version 1.2	2023-MAY-22

CONFIDENTIALITY/SECURITY WARNING

이 문서에 포함되어 있는 정보는 국민대학교 소프트웨어융합대학 소프트웨어학부 및 소프트웨어학부 개설 교과목 캡스톤 디자인 수강 학생 중 프로젝트 "지하철 커뮤니티 - 출그니"를 수행하는 팀 "28조"의 팀원들의 자산입니다. 국민대학교 소프트웨어학부 및 팀 "28조"의 팀원들의 서면 허락없이 사용되거나, 재가공 될 수 없습니다.

문서 정보 / 수정 내역


Filename	결과보고서-지하철 커뮤니티 - 출그니.doc
원안작성자	신명철
수정작성자	서민석, 송민준

수정날짜	대표수정자	Revision	추가/수정 항목	내 용
2023-03-25	신명철	1.0	최초 작성	문서 최초 작성
2023-05-20	송민준	1.1	수행 내용 작성	수행 내용 작성
2023-05-22	신명철	1.2	최종 검토	최종 검토

 국민대학교 소프트웨어학부 캡스톤 디자인 I	수행보고서		
	프로젝트 명	지하철 커뮤니티 - 출그니	
	팀 명	28조	
	Confidential Restricted	Version 1.2	2023-MAY-22

목 차

1	개요	4
2	연구/개발 내용 및 결과물	5
2.1	목표	5
2.2	연구/개발 내용	6
2.3	시스템 기능 요구 사항	8
2.4	시스템 비기능(품질) 요구 사항	9
2.5	시스템 구조 및 설계도	10
2.6	활용된 기술	10
2.7	결과물 목록	14
2.8	현실적 제한 요소 및 그 해결 방안	14
2.9	기대 효과 및 활용 방안	15
3	자기 평가	15
4	참고 문헌	16
5	부록	16
5.1	사용자 매뉴얼	16
5.2	운영자 매뉴얼	16
5.3	테스트 케이스	17

 <div> 국민대학교 소프트웨어학부 캡스톤 디자인 I </div>	수행보고서		
	프로젝트 명	지하철 커뮤니티 - 출그니	
	팀 명	28조	
	Confidential Restricted	Version 1.2	2023-MAY-22

1 개요

1. 프로젝트 개요

본 프로젝트는 Kotlin 언어로 개발된, Android Native 어플리케이션으로 지하철 커뮤니티 어플리케이션이다.

‘출그니’ 프로젝트는 ‘5’가지의 주요 기능을 제공한다. 아래의 5가지 주요 기능을 통해 지하철 이용에 있어 편리함을 제공하고자 한다.

- 커뮤니티

지하철에서 발생하는 실시간 상황이나, 정보를 유저들간 공유할 수 있는 SNS 서비스를 제공한다.

- 지하철 실시간 위치 정보

‘공공 데이터 포털’의 ‘서울특별시 지하철 실시간 위치정보’ API를 사용하여, 1호선부터 9호선까지 열차의 실시간 위치 정보를 제공한다.

- 지하철 도착 정보

‘공공 데이터 포털’의 ‘서울특별시 지하철 실시간 도착정보’ API를 사용하여, 1호선부터 9호선까지의 각 역사의 열차 도착 정보를 제공한다.

- 지하철 관련 정보


보다 편리한 지하철 이용을 위해, 지하철 관련 정보를 제공한다. 지하철 관련 정보는 크게 4가지가 제공된다.

1. 네이버 뉴스

“호선”, “지연” 등 지하철 관련 키워드를 가지고 네이버 뉴스 API에 요청하여 관련 뉴스를 전처리를 하여 ‘AWS Lambda’를 통하여 유저에게 제공한다.

2. 실시간 트위터 알림

서울 교통공사의 트위터 ‘Seoul metro’에서 제공하는 실시간 지하철 관련 트윗을 Twitter API를 이용하여 확인하고 이를 FCM을 이용하여 어플리케이션 알림을 유저들에게 전송하여, 실시간 정보를 제공한다.

 국민대학교 소프트웨어학부 캡스톤 디자인 I	수행보고서		
	프로젝트 명	지하철 커뮤니티 - 출그니	
	팀 명	28조	
	Confidential Restricted	Version 1.2	2023-MAY-22

3. 시위 정보

서울시 경찰청에서 제공하는 오늘의 시위정보를 유저들에게 제공한다.

4. 실시간 열차 트래픽

현재 트위터에서 가장 유동이 많은 순서대로 호선에 순위를 매긴다.

tweepy API에 요청하여 얻은 각각의 호선의 일정 기간 트윗된 수를 기준으로 순위를 매겨 유저에게 정보를 제공한다.

- 대중교통 길 찾기 기능

카카오 키워드 장소 검색 API를 이용하여, 출발지와 목적지를 설정하고, 설정된 출발지와 목적지를 바탕으로 SK 대중교통 API를 통해 최적의 경로를 제공한다. 경로의 경우에는 시간 순 또는 비용 순으로 설정이 가능하다.

2. 추진 배경 및 필요성

기존에 존재하는 지하철 애플리케이션은 "지하철 도착 정보", "노선도"에 초점을 둔 경우가 많다. 또한 지하철 실시간 정보를 얻기 위해서는 "서울시 교통 공사" 트위터를 통해 얻거나, 뉴스를 통해 얻어야 한다.

이러한 이유로 지하철의 지연이 발생했을 경우에 사전에 정보를 인지하지 못한다. 따라서 이러한 지하철 실시간 정보와 "지하철 노선도", "도착 정보"을 하나의 어플리케이션으로 해결할 수 있도록 "출그니"를 제작하게 되었다.

정보를 일방적으로 제공하는 것에서 벗어나 유저들이 직접 지하철 실시간 정보를 공유하며 더욱 생생하고 빠른 정보를 제공할 수 있도록 커뮤니티 기능을 중심으로 한 어플리케이션을 제작하고자 한다.

2 연구/개발 내용 및 결과물

2.1 목표

본 프로젝트의 목표는, 사용자가 보다 편리하게 정보를 얻기 위해 직관적인 UI와 기존 지하철 애플리케이션이 제공하는 필수적인 기능들을 제공하며 커뮤니티 기능이 존재하여 누구나 쉽고 빠르게 지하철 정보를 공유할 수 있는 어플리케이션을 제작하고자 한다.



- 실시간 시위 정보와, 대중교통 길찾기 서비스를 이용하여 사전에 시위를 인지하고 새로운 경로로 이동할 수 있다.
- 기습적 시위나, 지하철 관련 사고와 같이 사전에 인지하기 어려운 사고는 지하철 실시간 뉴스와 서울교통 공사 트윗 알림을 통해 인지할 수 있다.
- 타 지하철 어플리케이션이 기본적으로 제공하는, 지하철 도착 정보 및 위치 정보를 이용할 수 있다.
- SNS 기능을 통해 유저들간 지하철 정보를 공유하여 지하철 관련 정보를 얻을 수 있다.

2.2 연구/개발 내용

.2.2.1 JWT 토큰 인증

[\[캡스톤 중간보고서 - 2.2 수행내용\]](#) JWT 토큰 인증 방식으로 변경 페이지 참고

[\[기술 보고서 참고\]](#)

.2.2.2 로그인과 회원가입

[\[기술 보고서 참고\]](#)

.2.2.3 지하철 실시간 위치 정보

[\[캡스톤 중간보고서 - 2.2 수행내용\]](#) 실시간 열차 위치 정보 페이지 참고

[\[기술 보고서 참고\]](#)


.2.2.4 지하철 실시간 도착 정보

[\[캡스톤 중간보고서 - 2.2 수행내용\]](#) 실시간 도착 정보 페이지 참고

.2.2.5 대중교통 길찾기 기능

[\[캡스톤 중간보고서 - 2.2 수행내용\]](#) 대중교통 길찾기 페이지 참고

[\[기술 보고서 참고\]](#)

	수행보고서		
	프로젝트 명	지하철 커뮤니티 - 출그니	
	팀 명	28조	
	Confidential Restricted	Version 1.2	2023-MAY-22

.2.2.6 경찰청 시위 정보 가져오기

[\[기술 보고서 참고\]](#)

.2.2.7 Firebase를 이용한 Notification 만들기

[\[기술 보고서 참고\]](#)

.2.2.8 실시간 열차 트래픽 가져오기

[\[캡스톤 중간보고서 - 2.2 수행내용\]](#) Tweepy를 이용한 실시간 검색어 참고

[\[기술 보고서 참고\]](#)

.2.2.9 실시간 네이버 뉴스 가져오기

[\[기술 보고서 참고\]](#)

.2.2.10 Amazon S3 이미지 업로드

[\[캡스톤 중간보고서 - 2.2 수행내용\]](#) Amazon S3 이미지 업로드 페이지 참고

[\[기술 보고서 참고\]](#)

.2.2.11 요청에 대한 validation 수행, 응답 양식 변경


[\[기술 보고서 참고\]](#)

.2.2.12 이미지 파일을 Json 응답에 포함하기

[\[기술 보고서 참고\]](#)

.2.2.13 피드 페이지링 구현

[\[기술 보고서 참고\]](#)

 국민대학교 소프트웨어학부 캡스톤 디자인 I	수행보고서		
	프로젝트 명	지하철 커뮤니티 - 출그니	
	팀 명	28조	
	Confidential Restricted	Version 1.2	2023-MAY-22

.2.2.14 해시태그 필터링 구현

[\[기술 보고서 참고\]](#)

2.3 시스템 기능 요구 사항

시스템 기능 요구사항	내용	완료 여부
지하철 도착 정보	선택된 역사의 실시간 열차 도착 정보를 제공한다.	완료
지하철 열차 위치 정보	1호선부터 9호선까지 원하는 호선을 선택 가능하며, 각 호선 별 실시간 열차 위치 정보를 제공한다.	완료
로그인	로그인을 한다	완료
회원가입	회원이입을 한다	완료
대중교통 길찾기 서비스	입력된 출발지와 목적지를 기반으로 대중교통 길찾기 서비스(비용순, 시간순)를 제공한다.	완료
마이페이지	작성한 피드와, 팔로워 수, 작성한 피드 수, 팔로잉 수, 프로필 확인 및 수정할 수 있다.	완료
실시간 지하철 뉴스	지하철 관련 뉴스를 제공한다. 클릭 시 해당 뉴스 기사 브라우저로 이동한다.	완료
설정 기능	로그아웃 및 알림 기능 활성화를 선택할 수 있다.	완료
피드 작성	해시태그를 설정하고, 글과 사진을 첨부하여 피드를 작성한다.	완료
피드 확인	유저들이 작성한 피드를 최신 순으로 확인 가능하다. 필터링 기능을 통해, 원하는 태그의 게시글만 확인 할 수 있다. 피드 클릭 시 해당 유저의 정보를 확인 할 수 있다.	완료
피드 삭제	본인이 작성한 피드를 삭제한다 .	완료
댓글 기능	원하는 피드에 댓글을 작성할 수 있다.	완료



좋아요 기능	원하는 피드에 좋아요를 통해 감정표현을 할 수 있다. 한번 더 클릭 시 좋아요 취소가 가능하다.	완료
유저 검색	유저 닉네임 또는 유저 아이디를 통해 유저를 검색 가능하다. 검색된 유저를 선택 시 해당 유저의 정보를 확인 가능하다.	완료
유저 정보	해당 유저의 팔로우 수, 팔로잉 수, 작성한 게시물 수를 확인 가능하며, 해당 유저를 팔로우 할 수 있다.	완료
팔로우 기능	원하는 유저를 팔로우 할 수 있다.	완료
열차 트래픽 기능	1호선부터 9호선까지 일정 시간 기준으로 언급된 수와 이를 바탕으로 랭킹을 제공한다.	완료
오늘의 시위정보	오늘의 시위정보를 제공한다.	완료
서울 매크로 실시간 트윗	서울 매크로에 작성되는 실시간 트윗을 확인할 수 있다..	완료
지하철 노선 정보	1호선부터 9호선까지 지하철 노선 정보로 제공한다. 제공된 노선에는 열차의 실시간 위치 정보가 제공된다. 역사 클릭 시 열차 위치 정보를 제공한다.	완료
트위터 알림 기능	새롭게 트위터에 글이 올라올 경우 인앱 알림이 제공된다	완료

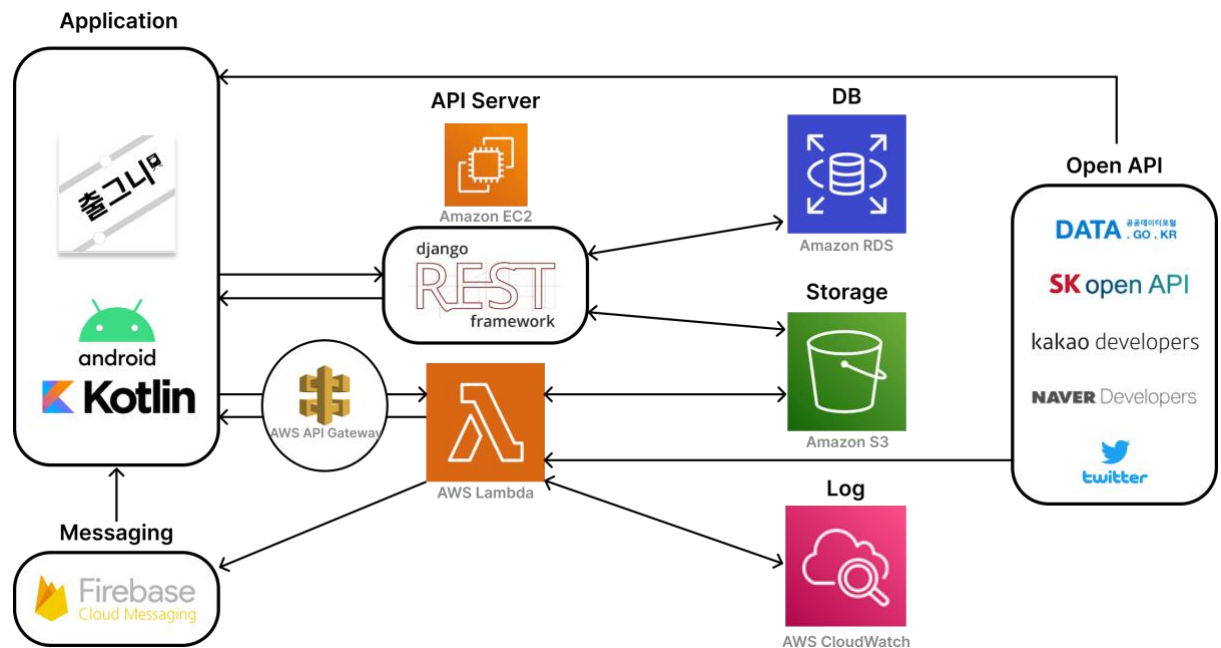
2.4 시스템 비기능(품질) 요구 사항

항목	설명	달성 여부
접근성	인터넷이 연결된 환경에서, 어플리케이션을 통해 사용할 수 있다.	달성
사용성	타 어플리케이션에서 사용하는 UI/UX를 사용하여 처음 사용하는 유저도 쉽게 사용할 수 있다.	달성
신뢰성	실시간으로 제공하는 어플리케이션이므로 오류 발생 확률이 낮아야 한다.	달성

 국민대학교 소프트웨어학부 캡스톤 디자인 I	수행보고서		
	프로젝트 명	지하철 커뮤니티 - 출그니	
	팀 명	28조	
	Confidential Restricted	Version 1.2	2023-MAY-22

효율성	API 서버의 응답 속도가 5초 이내여야 한다.	달성
-----	----------------------------	----

2.5 시스템 구조 및 설계도



2.6 활용된 기술

● 안드로이드

1. Glide 모듈

프로필 사진이나, 피드의 사진과 같이 해당 프로젝트에서는 이미지를 사용하는 경우가 많다. 효율성을 위해, 이미지를 빠르게 가져와야 하기에, 안드로이드에서 이미지를 빠르고 효율적으로 불러올 수 있는 Glide 라이브러리를 사용하였다.

2. Retrofit 모듈

서버와의 API 통신으로 받아온 데이터를 사용하기 위해, 필요한 형태 객체로 받



을 수 있는 Retrofit 라이브러리를 사용하였다. Retrofit 라이브러리는 OkHttp와 달리 연결 및 오류처리 캐싱과 같이 직접 해주어야 했던 작업들을 자동으로 잡아주기에 짧은 시간 내에 빠르게 개발할 수 있어 선택하였다.

3. Lottie 라이브러리

API 서버와의 통신과정에서 네트워크 상황에 따라 최대 3초 정도 소모가 된다. 통신 과정에서 화면을 빈 화면으로 두는 것이 아닌, 로딩화면 및 통신 실패의 결과에 대한 이미지를 보여주기 위해 Lottie 라이브러리를 사용하여, 애니메이션으로 직관적으로 현재 상황을 보여준다.

4. Image Crop

대부분의 SNS 어플리케이션의 경우 프로필을 설정할 때, 사용자에게 보여질 영역을 선택한다. 이를 제공하기 위해, Image Crop 라이브러리를 사용하였다.

5. Firebase FCM

실시간으로 알림을 보낼 수 있는 서비스로, 유저가 원하는 키워드의 주제를 구독을 하게 되면, 해당 주제의 구독자에게 원하는 메시지를 무료로 안정적으로 전송할 수 있는 Firebase에서 제공하는 서비스이다. 해당 서비스를 이용하여, 서울 매트로의 새로운 트윗이 작성이 되면 사용자에게 어플리케이션 알림 메시지를 전달한다.

6. ViewModel, LiveData

수명주기를 고려하여 데이터를 저장하고 관리한다. 예를 들어 안드로이드의 경우 화면이 회전과 같이 화면 변화가 발생하면 데이터가 변동될 수 있다. 이러한 상황을 막기 위해, ViewModel 객체를 이용하여 데이터를 저장하고 LiveData를 이용하여 데이터 변경 시 Observing하여 변경사항을 관찰 할 수 있다. UI와 컨트롤러를 분리하고, 액티비티와 프래그먼트 간 데이터 공유를 편리하게 하기 위해 사용하였다.



● 백엔드 (EC2 서버 측)

1. Django

API 서버를 구축하기 위한 목적으로 Django를 사용해서 유저로부터 오는 Select, Insert, Delete 요청에 대해 쿼리 결과를 알려주거나 데이터 베이스에 해당 요청의 파라미터를 통해 DB에 반영하고 다양한 비즈니스 로직을 수행하는 서버를 구축했다.

2. Django Rest Framework

어플에서 요청을 보내면 json으로 응답을 보내야한다. DRF는 Serializer를 통해서 파이썬 객체에서 json으로 변환하는 직렬화와 json에서 파이썬 객체로 변환하는 역직렬화를 수행한다. json을 다루는 요청, 응답에서 serializer가 필요했기 때문에 Django Rest Framework를 사용하게 되었다. serializer는 validation 또한 수행하기 때문에 올바른 요청을 보냈는지 검사하고 에러를 보낼 수 있다.

3. Json Web Token (JWT)

서버에서 회원가입, 로그인을 제외한 모든 요청은 JWT 인증 방식이 사용된다. JWT를 인증방식으로 사용함에 있어서 서버 측에 따로 토큰을 위한 데이터베이스가 없으며 파기 기한이 있어 다른 Base, Cookie 인증 방식보다 안전한 인증을 할 수 있다.

4. Boto3 (Amazon S3)

boto3는 아마존 서비스를 이용하기 위한 파이썬 라이브러리이다. 본 프로젝트는 사용자가 피드 이미지나 프로필 이미지를 S3에 업로드하고 가져온다. 유저가 서버에 이미지를 업로드하면 서버는 이미지 바이너리 코드를 가지고 boto3 클라이언트를 통해 s3에 이미지를 업로드한다. 또한 이미지를 요청할 때는 boto3 클라이언트를 통해 이미지를 받아서 utf8 형식으로 json에 넣어서 응답한다.

5. Amazon EC2

API 서버를 배포할 목적으로 EC2 인스턴스를 생성하고 프로젝트를 옮겼다. 탄력적 IP



를 생성하고 인스턴스에 연결했기 때문에 IP가 매번 변경되지 않아 클라이언트 측에서 IP를 변경하지 않아도 된다. EC2를 사용한 목적은 이후에 확장성 때문이기도 하다. Amazon EC2 Auto Scaling과 Elastic Load Balancing를 추가로 사용하여 트래픽 증가에 대한 대응으로 인스턴스 수를 증가시킴으로써 해결할 수 있기 때문이다. Auto Scaling과 Load Balancing은 아직 사용하지 않았다.

6. Amazon RDS

DB를 생성하기 위해 Amazon RDS를 이용했다. 데이터베이스 종류는 로컬에서 개발한 것과 같이 MySQL를 선택했다. EC2에서 RDS의 DNS를 통해 RDS의 데이터베이스에 접근하고 테이블 생성, 튜플 생성, 삭제등을 명령할 수 있다. RDS는 AWS에서 자체적으로 관리하는 서비스이기 때문에 확장성이나 가용성을 확보하는데 별다른 조치가 필요하지 않다.


● 백엔드 (Lambda 서버 측)

1. AWS Service

대부분의 서비스들은 AWS Service를 통하여 실행이 되며 Lambda, CloudWatch, EC2, IAM 와 같은 서비스들을 이용하여 코드를 구현하고 관리하며 이를 통하여 보안을 중요시할 수 있고 플랫폼 문제도 해결이 되며 어디서든 사용할 수 있는 장점들이 존재한다. 이러한 점들을 통하여 개발자들과 코드 공유도 쉽게 가능하며 프론트와 백엔드가 API 게이트웨이를 통해 연결이 되며 작업을 구현해 나간다.

2. Firebase FCM

사용자에게 알림을 주기 위해서는 Firebase FCM을 이용하면 구독한 사용자에게 알림을 보내는 서비스를 생성할 수 있다. 이는 Python을 이용하여 구현이 가능하며 원하는 코드를 실행하고 만약에 특정 목적을 달성했을 경우 Firebase FCM

 국민대학교 소프트웨어학부 캡스톤 디자인 I	수행보고서		
	프로젝트 명	지하철 커뮤니티 - 출그니	
	팀 명	28조	
	Confidential Restricted	Version 1.2	2023-MAY-22

을 통하여 구독을 한 사용자에게 알림을 보내는 형태이다.

2.7 결과물 목록

- [어플리케이션 다운로드 링크](#)



- 어플리케이션 시연 영상

[\[시연영상\]](#)

- 중간 / 기말 발표 자료 및 보고서

[\[중간/기말 발표자료 및 보고서\]](#)

링크 : <https://github.com/kookmin-sw/capstone-2023-28/tree/master/docs>

2.8 현실적 제한 요소 및 그 해결 방안

대중교통의 경우에는 서울 이외에도 부산, 대구, 대전과 같은 도시에도 존재한다. 한국 내 모든 지하철 노선에 대한 정보를 제공하는 것은 캡스톤 단계에서 모든 데이터를 가공하고 관리하는 것이 어렵기에, 수도권 지하철 노선인 1호선부터 9호선까지의 노선만 제공한다.

추후 진행할 개발에서는 전국 모든 노선에 대한 노선 정보를 제공하도록 할 예정이다.

2.9 기대 효과 및 활용 방안

하나의 어플리케이션 안에서 지하철 관련 정보를 얻을 수 있다는 점에서, 지하철을 이용하는 학생, 직장인 등 사람들에게 편리함을 제공할 수 있다.

기존의 존재하던 타 지하철 어플리케이션은 정보를 일방적으로 제공을 받았더라면, 본 어플리케이션은 커뮤니티 기능을 도입함으로써, 유저들 간 양방향으로 정보를 공유할 수 있다. 이를 통해 보다 빠른 실시간 정보를 얻을 수 있다.

커뮤니티 활성화가 이루어진다면, 지하철 상권에 대한 홍보 게시물 작성을 통해 지하철 상권 활성화를 기대할 수 있다.

3 자기 평가

성명	자기평가
신명철	백엔드 파트의 팀원들과 함께 개발을 하면서, 백엔드 파트에서의 과정의 이해도 필요로 하여 팀원과 많은 소통을 진행하였다. 소통을 통해 어느 정도 흐름을 이해할 수 있었고, 이해를 바탕으로 요구사항을 더 세부적으로 전달할 수 있었다. 프로젝트의 결과로 단순히 안드로이드 개발 부분에 있어서의 성장도 있었지만, 관련 자료를 조사하고 조장으로서 팀원들을 관리하며 다른 팀원의 파트도 이해하려는 노력을 통해 여러 부분에서의 성장이 있었다고 생각한다.
서민석	다양한 서비스를 이용해보며 혼자서 개발하는 것이 아닌 여럿이서 개발하는 것이 신기한 체험이 되었으며 부족한 부분이 많았지만 서로 회의하고 결과를 도출하는 프로세스를 통해서 많이 발전할 수 있었으며 애플리케이션 개발에 집중할 수 있었습니다.
송민준	애플리케이션 설계부터 개발 완료까지 체계적으로 진행한 적은 처음이라 신선한 경험이 되었습니다. 백엔드 서버를 개발하면서 생각보다 비즈니스 로직 안에 고려해야 될 점이 많았고 생각치 못한 오류를 여러 번 겪으며 개발자로서 성장을 했던 것 같습니다.

4 참고 문헌

번호	종류	제목	출처	발행년도	저자	기타
1	기술문서	Glide 공식 문서 https://github.com/bumptech/glide				
2	기술문서	Retrofit 공식 문서 https://square.github.io/retrofit/				

5 부록

5.1 사용자 매뉴얼

개인 스마트폰

- [다운로드 링크](#)

1. 링크 접속 후 APK 파일 다운로드
2. 출처를 알 수 없는 앱 설치 허용

- 어플리케이션 사용 방법


[\[시연영상\] 참고](#)

5.2 운영자 매뉴얼

- DB 설치 방법

[README.md](#) 페이지 참고

- AWS Lambda 사용 방법

	국민대학교 소프트웨어학부 캡스톤 디자인 I	수행보고서		
		프로젝트 명	지하철 커뮤니티 - 출그니	
		팀 명	28조	
		Confidential Restricted	Version 1.2	2023-MAY-22

[README.md](#) 페이지 참고

5.3 테스트 케이스

대분류	소분류	기능	테스트 방법	기대 결과	테스트 결과
로그인	회원가입	새로운 유저를 생성한다	회원 가입 버튼을 누르면, 입력한 email, password, 닉네임에 해당하는 유저를 생성한다. 이 때, 중복 이메일의 경우에는 생성되지 않는다.	새로운 유저를 생성한다.	성공
-	로그인	기존 유저일 경우 Token을 받아온다	회원가입 된 email 과 password를 입력하고 login 버튼을 누른다.	로그인이 되어 홈으로 이동되며, REST API 통신에 필요한 토큰을 받아온다.	성공
-	로그아웃	로그인 된 계정을 로그아웃한다.	하단 네비게이션 바에 My 버튼을 눌러 My 화면으로 이동 후 설정 버튼을 누르고 로그아웃 버튼을 누른다.	로그아웃이 되어, 로그인 화면으로 이동한다	성공
피드	피드 작성	글과 사진을 자신이 선택한 필터 태그와 함께 피드를 작성한다.	홈 화면에서 '지금 지하철은'을 클릭하여, 원하는 내용의 글을 작성하고, 사진을 촬영하거나, 갤러리에서 사진을 선택한다. 원하는 필터링 태그를 선택한 후에 작성하기를 누른다.	작성된 글과 태그는 DB에 저장되며, 사진은 S3에 저장되고 저장된 경로가 DB에 저장된다.	성공
-	피드 삭제	자신이 작성한 피드를 삭제한다.	자신이 작성한 피드의 경우 피드에 휴지통 모양의 삭제 아이콘이 있다. 해당 버튼을 클릭 시 삭제 확인을 물어보는 다이얼로그가 나타나고 삭제 버튼을 누르면 삭제가 된다	DB에서 해당 피드와 관련된 정보가 삭제가 된다.	성공



-	피드 좋아요	피드의 좋아요를 한다	피드의 좋아요 버튼을 누르면 좋아요를 표시할 수 있다, 좋아요가 눌린 상태에서 한번 더 클릭시 취소가 된다.	해당 피드에 좋아요 개수가 1 증가한다	성공
-	피드 목록	최신 순으로 피드의 목록이 나타난다	화면의 하단 네비게이션 바에서 피드를 선택하게 되면, 피드의 목록이 나타난다, 하단 스크롤시 새로운 피드가 업데이트 된다.	피드의 목록이 나타난다. 하단 스크롤시 새로운 피드가 업데이트가 된다.	성공
-	피드 새로고침	피드의 목록을 새로고침 한다	피드 화면에서 최상단에서 아래로 화면을 당기면 새롭게 피드가 업데이트가 된다.	새로운 피드로 업데이트가 된다	성공
-	피드 필터링	원하는 태그가 포함된 피드를 볼 수 있다	상단의 태그 버튼을 누르고, 자신이 원하는 태그를 선택 후 적용하기를 누르면 원하는 태그가 포함된 피드가 화면에 업데이트 된다.	자신이 선택한 태그가 포함된 피드가 화면에 업데이트 된다	성공
필터링	필터링 초기화	선택된 필터를 초기화 한다.	상단의 태그 버튼을 누르고, 초기화 버튼을 누르면 선택된 필터가 초기화 된다.	필터가 초기화 된다	성공
-	필터링 적용	선택된 필터를 적용한다.	필터 적용하기 버튼을 클릭한다	필터링이 적용된다.	성공
프로필	내 프로필	나의 프로필을 확인한다.	하단 네비게이션 바에, My 버튼을 클릭한다..	나의 작성한 게시글 수, 팔로우 수, 팔로잉 수, 작성한 글과 나의 이름과 상태 메시지가 화면에 나타난다.	성공
	내 프로필 수정	자신의 프로필을 수정한다.	프로필 수정 버튼을 클릭한다. 또는 사진을 클릭하여, 프로필만 별도로 바꿀 수 있다. 프로필과, 닉네임, 상태메시지를 바꿀 수 있으며, 원하는 정보를 입력 후 적용 버튼을 누르면 된다.	입력된 프로필 정보가 바뀐다.	성공
	다른 유저의	다른 유저의	유저 검색, 다른 유저의 피드를 클릭	해당 유저의 게시글 수, 팔로잉 수, 팔로우	성공



	프로필	프로필이다.	시 확인 가능하다.	수, 작성한 피드 목록과 팔로우 기능 버튼이 화면에 나타난다.	
팔로우	팔로우	다른 유저를 팔로우 한다.	팔로우를 원하는 유저의 프로필로 이동 후 팔로우 버튼을 누른다.	팔로우가 완료된다.	성공
	언팔로우	다른 유저를 언팔로우 한다.	언팔로우를 원하는 유저의 프로필로 이동 후 언팔로우 버튼을 누른다. 단 해당 유저를 팔로우한 상태여야 가능하다.	언팔로우가 완료된다	성공
댓글	댓글 작성	피드에 댓글을 작성한다.	피드의 메세지 아이콘을 클릭 후 원하는 댓글을 작성 후 작성 완료 아이콘을 누른다.	댓글 작성이 완료되어, 화면에 업데이트 된다.	성공
지하철	지하철 노선도	지하철 노선도를 확인 가능하다.	하단 네비게이션 바에서 노선도를 클릭 후 이동하여, 자신이 원하는 노선을 하단의 1~9호선 까지 선택이 가능하다.	자신이 선택한 노선에 해당하는 노선 정보와 실시간 열차위치가 화면에 나타난다.	성공
	지하철 실시간 위치 정보	열차의 실시간 위치 정보를 확인 가능하다.	지하철 노선도에서 원하는 노선을 클릭한다.	자신이 선택한 노선의 실시간 열차위치가 나타난다.	성공
	열차 도착 정보	해당 역사에 열차 도착정보를 제공한다.	상단의 역사 검색 또는 지하철 노선도의 원하는 역사를 선택하면 열차 도착 정보 화면으로 이동한다.	해당 역사에 도착 예정인 열차 정보가 화면에 나타난다.	성공
	역사 검색	역사를 검색한다	상단에 역사 검색 버튼을 누른 후 검색하고자 하는 역사를 입력한다.	자신이 입력한 결과에 맞는 역사가 화면에 나타난다	성공
길찾기	대중교통 길찾기	대중교통을 이용한 길찾기 기능이다.	자신이 원하는 출발지와 목적지를 입력이 완료되면, 해당 경로의 길찾기 정보가 나타난다. 길찾기 정보의 결과로는 버스/지하철/버스+지하철 총 3개의 결과로 선택이 가능하며, 시간	자신이 입력한 경로에 맞는 결과가 화면에 나타난다.	성공



			순 비용 순으로 나열이 가능하다.		
실시간 정보	실시간 뉴스	지하철 관련 뉴스를 제공한 다.	하단 네비게이션의 '오늘' 탭을 눌러 '오늘' 탭으로 이동한다.	로딩이 완료되면 지하 철 관련 뉴스가 화면 에 나타난다.	성공
	서울 교통 공사 실시간 트윗	서울 교통공사 의 실시간 상 위 5개의 트윗 을 제공한다.	하단 네비게이션의 '오늘' 탭을 눌러 '오늘' 탭으로 이동한다.	로딩이 완료되면 서울 교통공사 실시간 트윗 이 화면에 나타난다.	성공
	오늘의 시위 정보	오늘의 시위 정보를 제공한 다.	하단 네비게이션의 '오늘' 탭을 눌러 '오늘' 탭으로 이동한다.	로딩이 완료되면 오늘 의 시위정보가 제공된 다	성공
	실시간 열차 트래픽	실시간 열차 트래픽 정보가 제공된다.	하단 네비게이션의 '오늘' 탭을 눌러 '오늘' 탭으로 이동한다.	로딩이 완료되면 오늘 의 실시간 열차 트래 픽이 제공된다.	성공
알림	실시간 알림	교통 공사에서 새로운 정보가 제공되면 알림 이 온다	Firebase FCM을 통해 메시지를 전송 한다.	FCM을 통해 설정한 메세지의 포맷으로 기 기의 알림이 온다.	성공