

정에스더 | 문영민

# 부산 하철 프로젝트

BUSAN SUBWAY PROJECT

운행 시작합니다.





# Contents

1. 개요(Overview)
2. 팀 소개(Team Introduction)
3. 기술 스택(Tech Stack)
4. ERD
5. 세부 기능(Features)
6. 아키텍처(Architecture)
7. Development



# Overview



팀명 : BSP



프로젝트명

: 부산 지하철 프로젝트(Busan Subway Project)



프로젝트 제작 기간

: 2024.06.11.(화) ~ 06.27.(목)



개발 인원 : 2명



요약 : 지하철 운행 정보 앱



대상 : 부산 지하철을 이용하는 모든 승객



# Overview



목적

- 하루 평균 185만 명 이상의 승객이 이용하는 부산 지하철의 운행 정보를 보다 신속하게 제공
- 수도권 지하철 앱 수에 비해 현저히 떨어지는 부산 지하철 앱 수를 보완
- 즉각적인 현재 위치 및 역 이름 파악
- 타이머 기능 제공을 통해 효율적인 승하차 유도 및 시간 관리



# Team Introduction

문영민



@Moon-ymin



- **Front-End**
- **Back-End**
- **Database**

## Front-End

### 안드로이드(Android)\_메인 화면 구현

- 줌 인/줌 아웃 및 좌우 이동
- 노선도 이미지 좌표
- 한/영 스위치
- 검색 - 서치 뷰(SearchView)
- 경로 선택 팝업 윈도우
- 경로 찾기 - 카드 뷰(CardView)
- 액티비티 간 인텐트
- 사용자 설정 시간에 따른 열차 시간 변경

## Back-End

### 스프링 부트(SpringBoot)

- 스프링 부트 - MySQL 및 SQLite 연동
- 스프링 부트 - 안드로이드 연동 및 데이터 교환
- 노선도 좌표 클릭 시 역 이름 반환
- 최단 시간 및 최소 환승 구현 → 다익스트라 알고리즘

### 데이터베이스(Database)

- 파이썬(Python) 크롤링 및 파싱
- 공공 데이터 API 활용
- 데이터 입력

yet Humetro Ticket Hume



# Team Introduction

정에스더 @EstherOVO



- **Front-End**
- **Back-End**
- **Database**

## Front-End

### 안드로이드(Android)\_경로 확인 화면 구현

- 최단 시간 및 최소 환승 탭 레이아웃 (TabLayout)
- 메인으로 돌아가기 버튼
- 총 소요 시간 및 역 개수 화면
- 사용자 시간 설정
- 타이머
- 출발 | 경유 | 도착 화면
- 경유역 리사이클러 뷰(RecyclerView)
- 액티비티 간 인텐트
- 사용자 설정 시간에 따른 열차 시간 변경

## Back-End

### 스프링 부트(SpringBoot)

- 스프링 부트 - MySQL 연동
- 모델(Model) 및 테이블 생성

### 데이터베이스(Database)

- DB 설계
- Hadoop - MySQL 연동
  - 공공 데이터 파싱
  - 데이터 전달(Entity / Mapper / Driver 활용)
- 데이터 입력

jet Humetro Ticket Humetto Database Hume



# Tech Stack

Languages



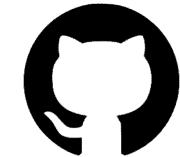
Front-End



Back-End



Version Control



Database

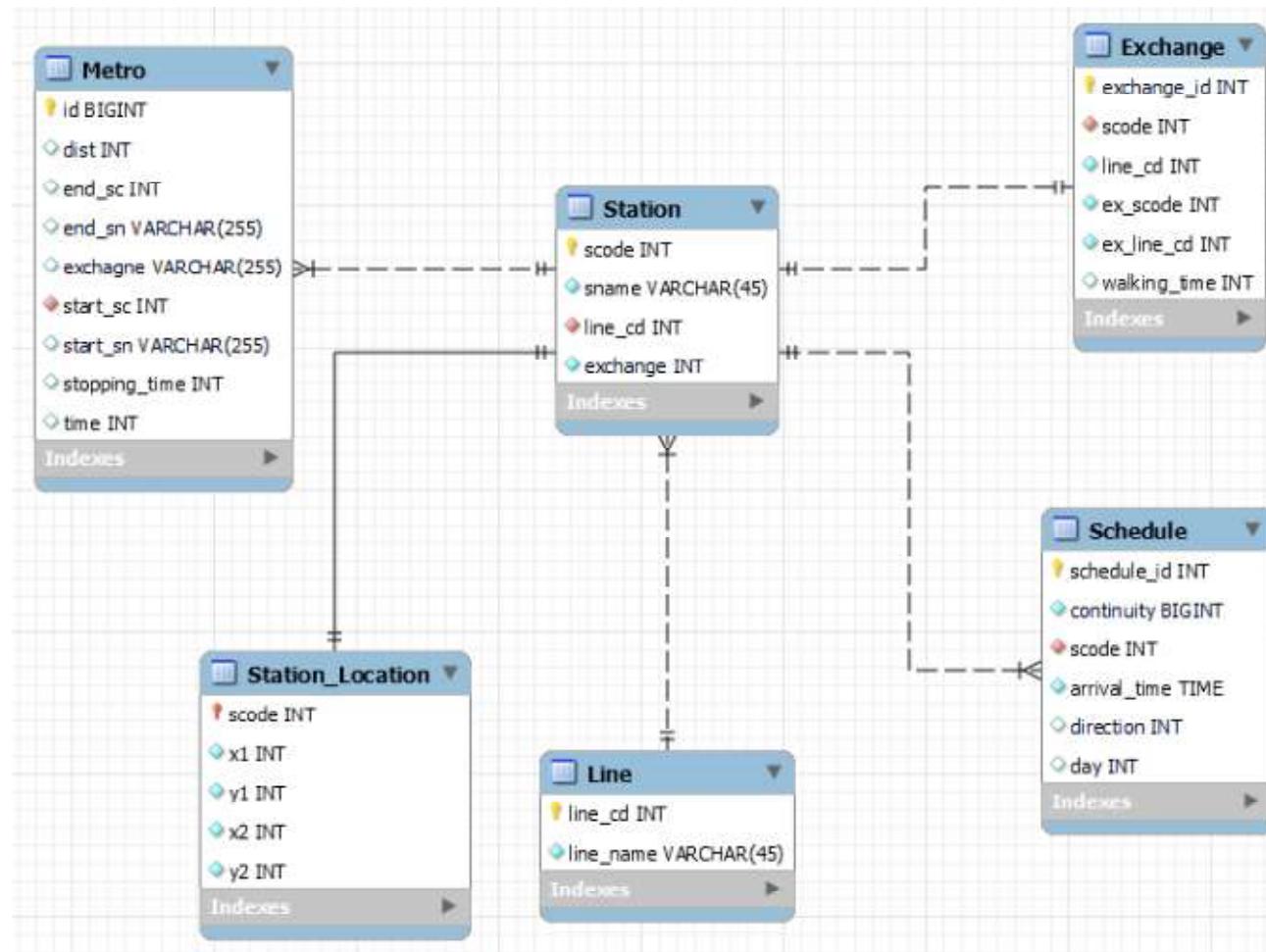


Design Tools





# ERD(Entity-Relationship Diagram)





# Features

## 1. 부산 지하철 노선도

- 국문 및 영문 노선도 제공

## 2. 출발 / 경유 / 도착 버튼 설정 시 최단 소요 시간

### 및 최소 환승 안내

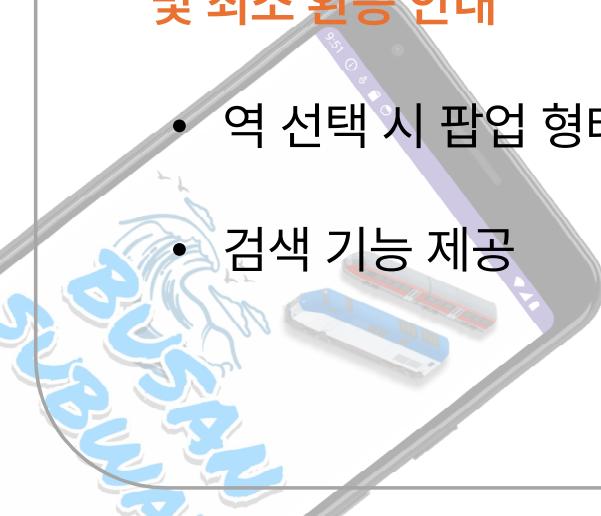
- 역 선택 시 팝업 형태로 알림
- 검색 기능 제공

## 3. 도착 예정 시간 기반 타이머 제공

- 별도의 알림 버튼을 클릭하여 타이머 시작
- 도착 예정 시간에 맞춰 알림 및 알람

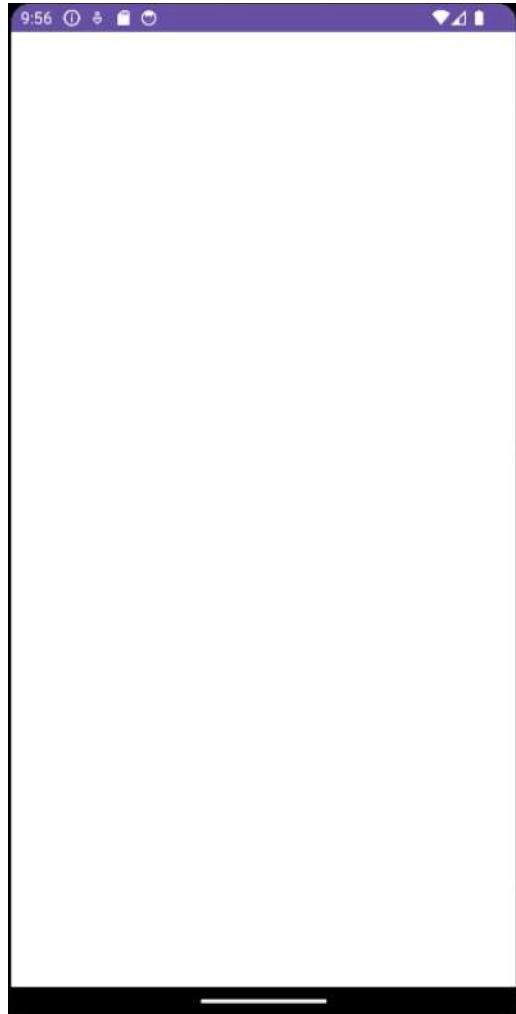
## 4. 영어 서비스 지원

- 외국인 승객을 위한 영어 서비스 지원

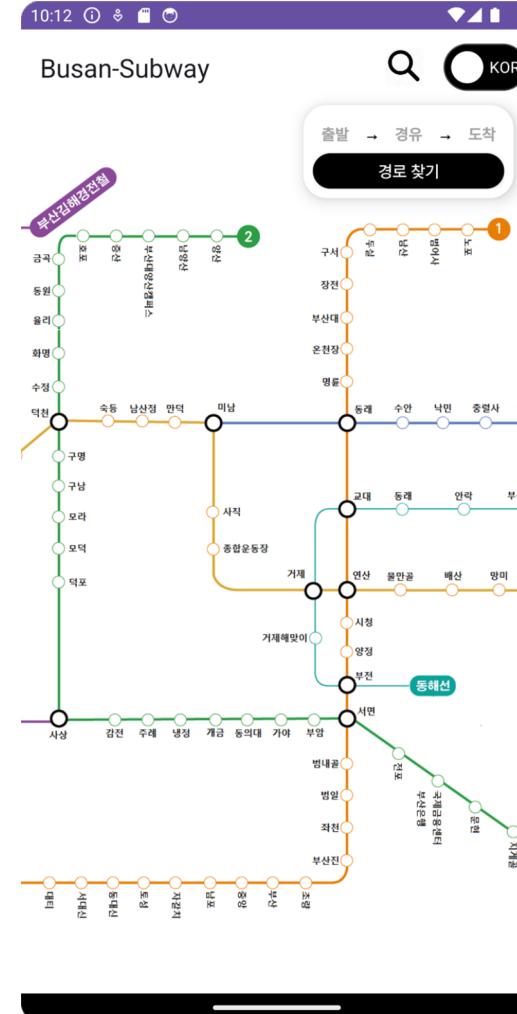




# Features



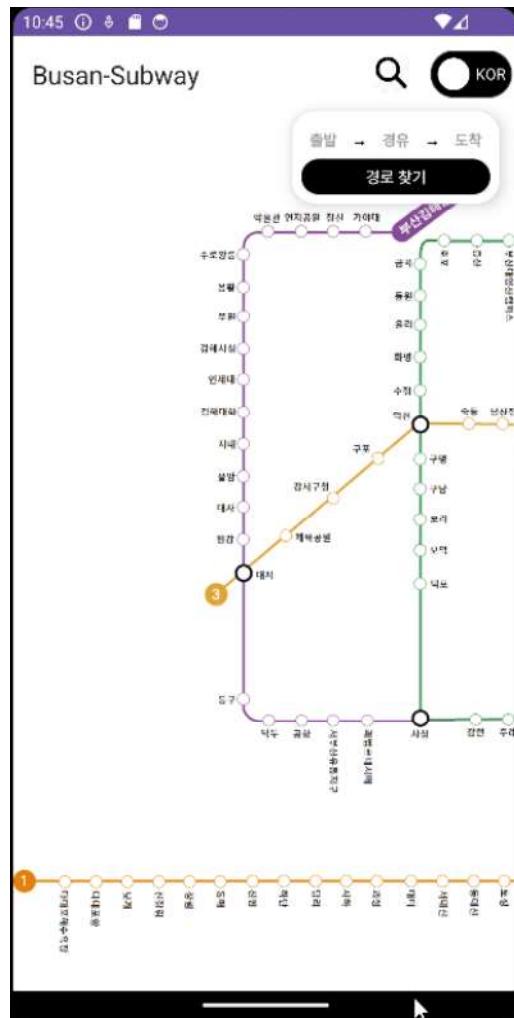
» 대기 화면



메인 화면(한) <<

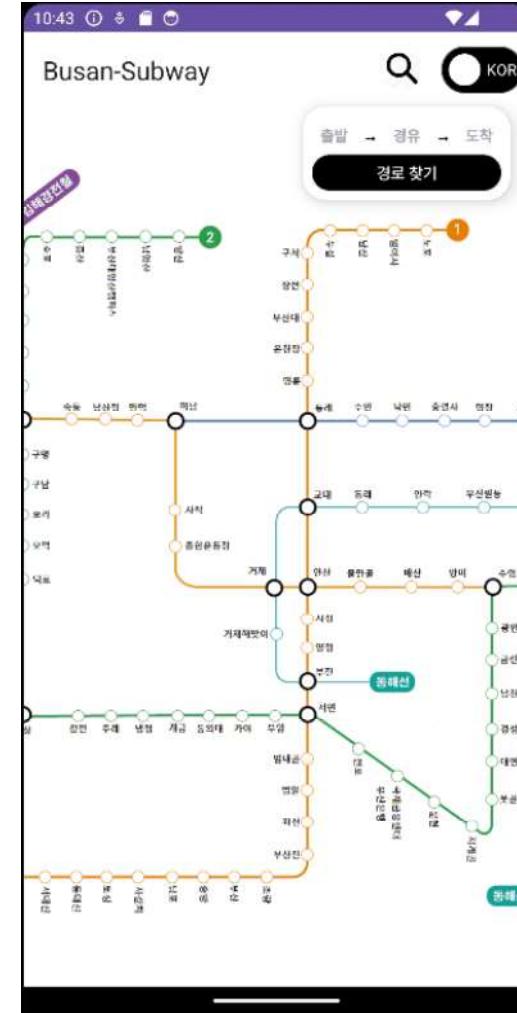


# Features



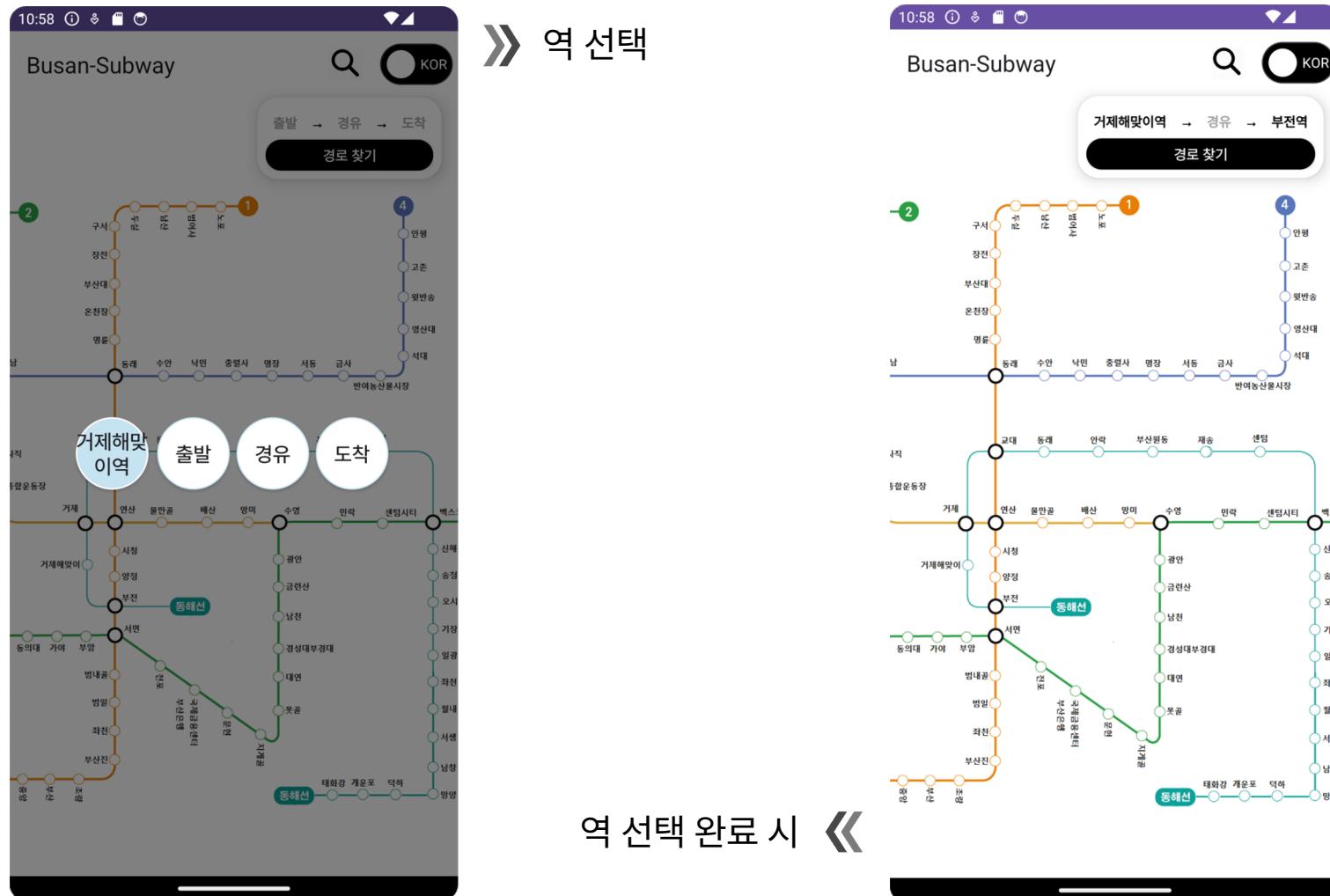
» 좌우 화면 이동

줌인 / 줌아웃 <>





# Features





# Features

11:03 ⓘ 🔍 🗃

Busan-Subway

최단시간

최소환승

약 8분 3개 역 이동

출발 시간 설정 ↗

타이머 설정 ⏳

동 거제해맞이역

지금 가장 빠른 열차는 11:31



① 부전역

도착 예정 시간은 11:40



메인으로 돌아가기

» 운행 확인 화면

11:04 ⓘ 🔍 🗃

Busan-Subway

최단시간

최소환승

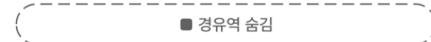
약 8분 3개 역 이동

출발 시간 설정 ↗

타이머 설정 ⏳

동 거제해맞이역

지금 가장 빠른 열차는 11:31



동 부전역

11:35:00

① 부전역

도착 예정 시간은 11:40

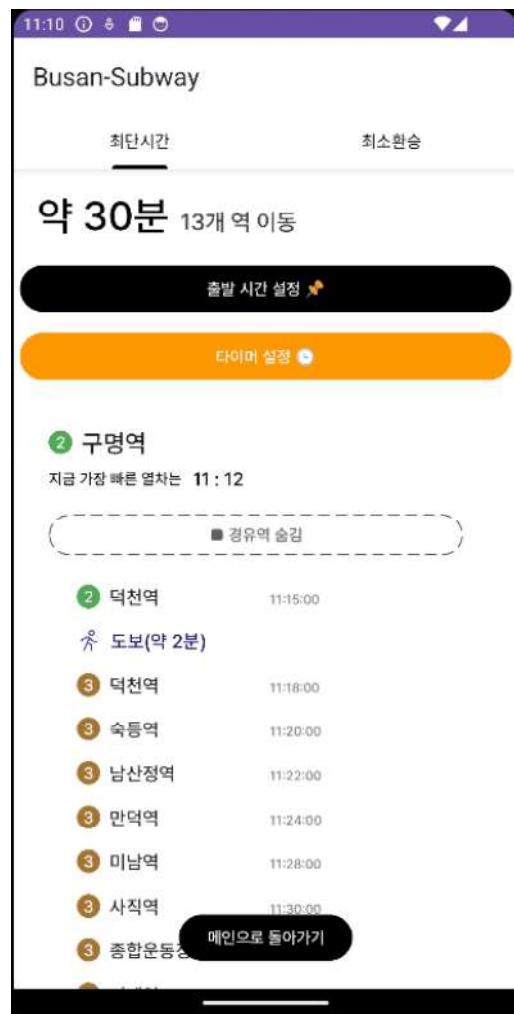


경유역 확인 ⇤

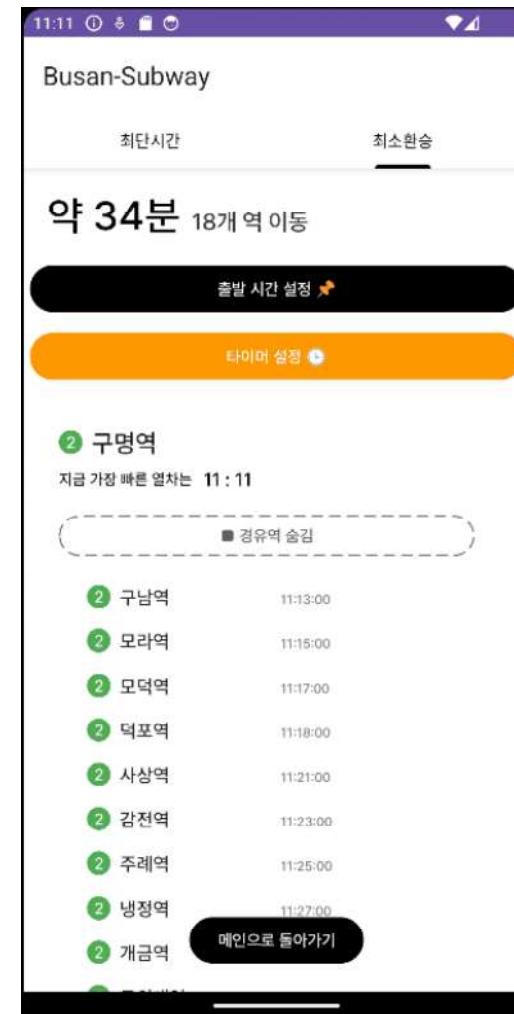
메인으로 돌아가기



# Features



» 최단 시간



최소 환승 <<



# Features



Busan-Subway

최단시간

최소환승

약 9분 6개 역 이동

출발 시간 설정 ↗

타이머 설정 ⏳

② 감전역

지금 가장 빠른 열차는 11:23



② 주례역

11:25:00

② 냉정역

11:27:00

② 개금역

11:29:00

② 동의대역

11:30:00

② 가야역

도착 예정 시간은 11:32



메인으로 돌아가기

» 가장 빠른 열차



최단시간

최소환승

소요시간

9분 환승없음 | 카드 1,600원

출발 오전 11:23 ~

도착 오전 11:32 ~

11:23 2 감전 &gt;

장산행 | 빠른 하차 5-1 &gt;

5개 역 (8분) ⇩

11:32 하차 가야 &gt;

내리는문 오른쪽 | 출구·버스정보 &gt;

▶ 톡 공유

▶ 경로 더보기

잘못된 정보 신고 &gt;



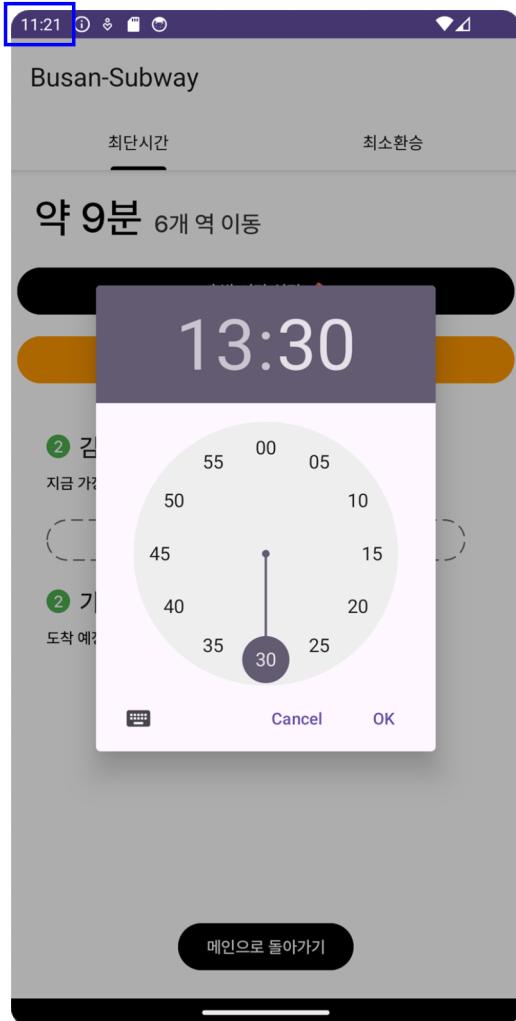
카카오 지하철 ◀

2024년 12월 31일까지  
100톤 이상 기존화학물질 등록하세요

i



# Features



» 사용자 시간 설정



카카오 지하철 <<



# Features



Busan-Subway

최단시간

최소환승

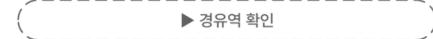
약 9분 6개 역 이동

출발 시간 설정 ↗

타이머 설정 ⏳

② 감전역

지금 가장 빠른 열차는 13 : 36



② 가야역

도착 예정 시간은 13 : 45

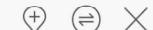


메인으로 돌아가기

» 가장 빠른 열차



최단시간 최소환승



소요시간

**9분** 환승없음 | 카드 1,600원

출발 오후 01:36 ▾

도착 오후 01:45 ▾

01:36 ↗ 2 감전 >

장산행 | 빠른 하차 5-1 >

5개 역 (8분) ⇩

01:45 하차 ↘

가야 >

내리는문 오른쪽 | 출구·버스정보 >

▶ 톡 공유

▶ 경로 더보기

잘못된 정보 신고 >



무신사 앱에서 확인!  
단 24시간, 오늘만 이 가격!  
무신사 반팔셔츠 한장에 27,400원!



i

카카오 지하철 ◀



# Features

11:35 ⓘ 🔍 🗃 🔍

Busan-Subway

» 타이머

최단시간

최소환승

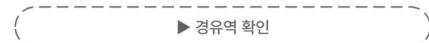
약 9분 6개 역 이동

출발 시간 설정 ↗

00 : 10 : 53

② 감전역

지금 가장 빠른 열차는 11 : 37



② 가야역

도착 예정 시간은 11 : 46



11:36 ⓘ 🔍 🗃 🔍

Busan-Subway

최단시간

최소환승

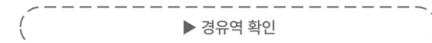
약 9분 6개 역 이동

출발 시간 설정 ↗

02 : 08 : 51

② 감전역

지금 가장 빠른 열차는 13 : 36



② 가야역

도착 예정 시간은 13 : 45



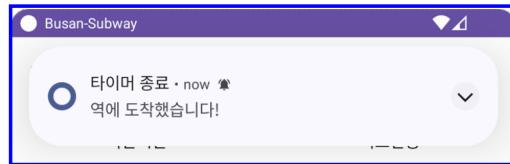
메인으로 돌아가기

사용자 설정 타이머 <<

메인으로 돌아가기



# Features



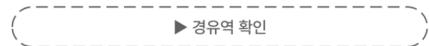
약 9분 6개 역 이동

출발 시간 설정 ❤

00 : 00 : 00

② 감전역

지금 가장 빠른 열차는 11 : 51



② 가야역

도착 예정 시간은 12 : 00

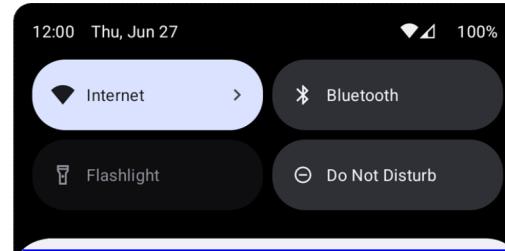


\* 타이머 시작 전 시간 → 11 : 46

11:46 ⓘ 🔍 🌐

메인으로 돌아가기

## » 토스트 메시지 알림



Busan-Subway • now ⓘ

타이머 알림

타이머가 종료됐습니다. 시간을 확인하세요!

Busan-Subway • now ⓘ

역에 도착했습니다!

New ads privacy features now available • 2d

Now, there's more choice over the ads you see...

Silent

Need time to focus?

Use Focus mode to pause distracting apps. T...

Virtual SD card

For storing photos, videos, music and more

Android System

Serial console enabled. Performance is impacted. T...

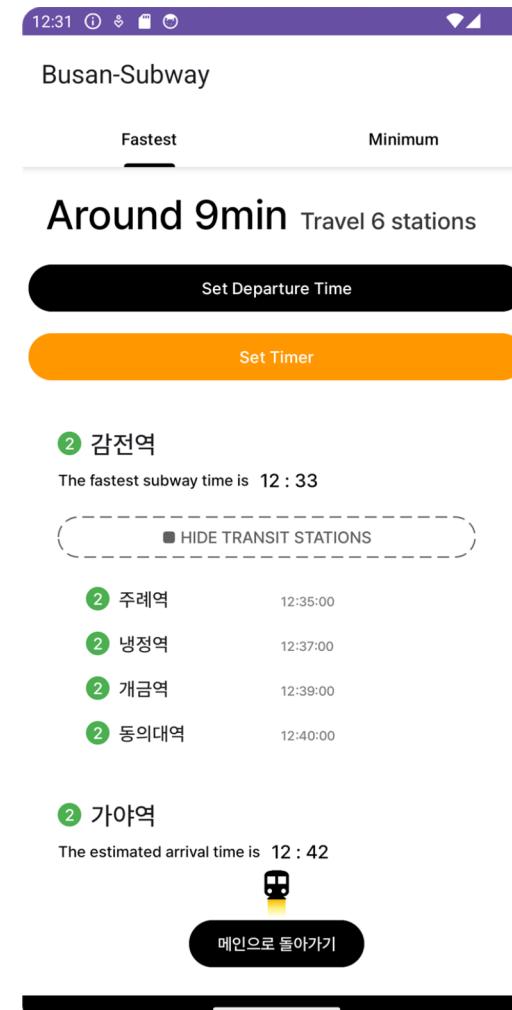
알람 ↵



# Features



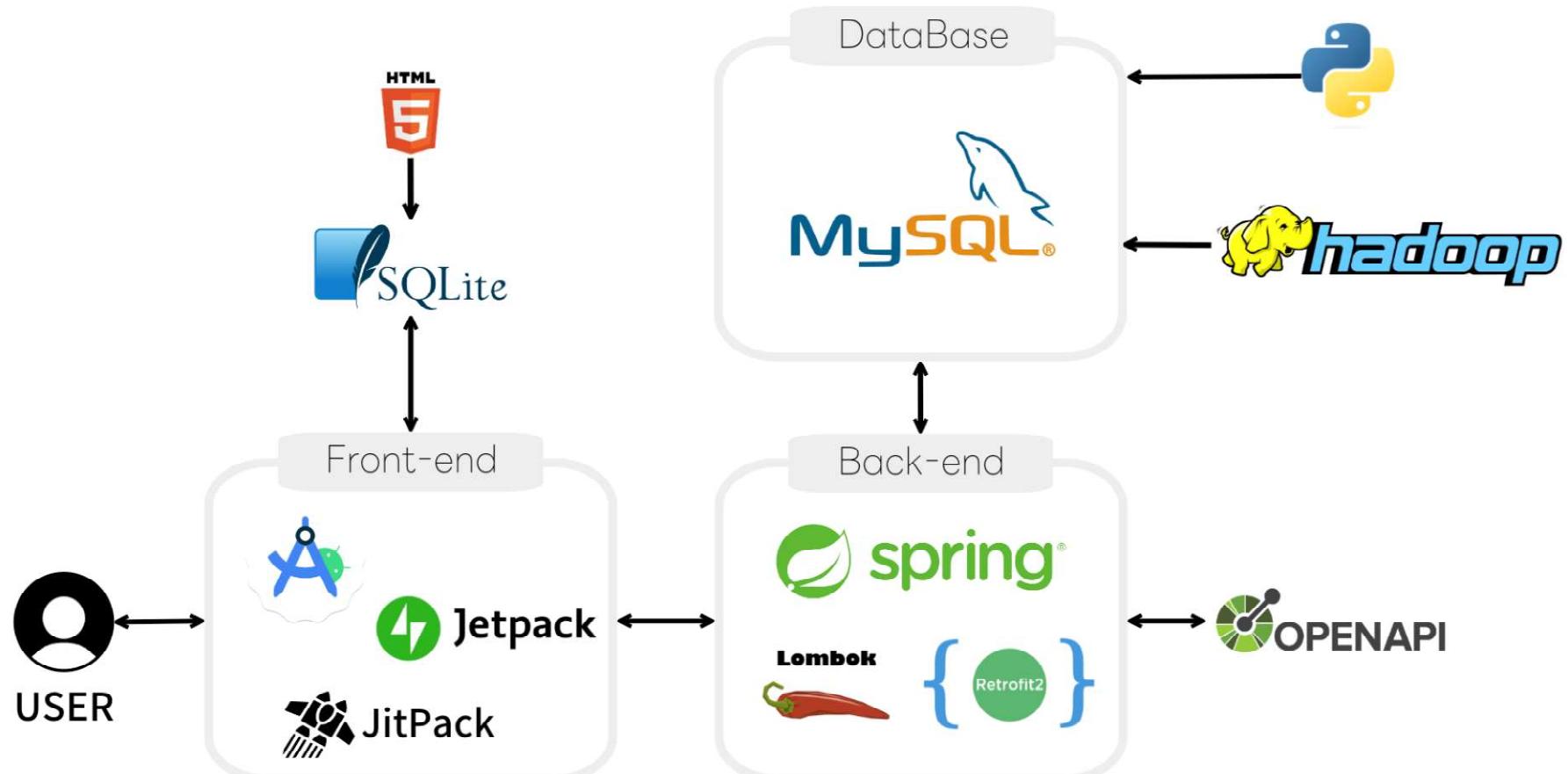
» 영어 버전(메인)



영어 버전(결과) ◀



# Architecture





# YoungMin's Development

## 개인

(FE) 메인 화면 좌표를 구하는 데 어려움이 있었다.

에뮬레이터 화면에 맞춰 좌표를 구하고 SQLite에 즉각 저장하여 해결!

(FE) 운행 시간표 중 도착 시간이 없고 출발 시간만 나와 있는 데이터가 있어서 크롤링 하는 데 한계가 있었다.

홈페이지에 나와 있는 지하철 운행 시간표를 바탕으로 데이터를 직접 입력하여 해결!

(BE) 최단 시간 알고리즘은 다익스트라, 최소 환승 알고리즘은 BFS를 주로 사용한다고 하는데, 알고리즘 설계가 어려웠다.

역 간 연결 관계가 많은 프로젝트 특성상 최단 시간, 최소 환승 모두 다익스트라 알고리즘으로 해결!



FE : 프론트엔드에서 어려웠던 점  
BE : 백엔드에서 어려웠던 점

## 팀(공통)

(BE) 사용자가 직접 시간을 설정할 때 역직렬화 문제로 LocalTime으로 전달된 데이터가 올바르게 처리되지 못했다.

입력된 데이터를 String으로 변환한 뒤, 요구하는 데이터 형태에 맞게 서버에서 다시 LocalTime 객체로 변환하여 해결!

(BE) 위의 연장선으로, 사용자가 시간 설정에 따라 가장 빠른 열차 시간이 바뀌어야 하는데, 이를 설계하는 데 어려움이 있었다.

메인에서 다른 액티비티로 넘어갈 때 입력 역 정보도 같이 전달 → 시간이 바뀔 경우 Fragment에서 서버로 바로 연결하여 변경된 정보 및 데이터를 교환하여 해결!

(BE) 수업 중 다뤘던 Spring Security를 회원 가입 및 유저 정보 저장 등으로 프로젝트에도 적용하고 싶었으나 그러지 못해서 아쉬웠다.

시간 부족의 한계임을 인정하고 다음에 더욱 상세하게 다룰 수 있도록 개인적으로 계속 공부를 할 예정!



# Esther's Development

## 개인

(FE) 예상 소요 시간에 따라 타이머가 설정되고, 타이머 종료 후 알림 및 알람을 구현하는 데 어려움이 있었다.

타이머는 TimePickerDialog 및 텍스트 수정, 알림 및 알람은 AlarmReceiver를 구현하여 해결!

(FE) 서버에서 전달받은 데이터를 사용자 화면에 나타내는 과정에 난관이 있었다.

메인에서 인텐트(Intent)를 구현해 다른 액티비티와 연결해 데이터를 넘겼으며, 이를 메서드 별 요구하는 데이터 특성에 맞게 활용하여 해결!

(BE) 데이터를 파싱하고 이를 Hadoop과 MySQL을 연결한 뒤 전달하는 과정이 어려웠다.

테이블이 가진 컬럼의 특성(PK 및 FK 등)을 이해한 뒤 요구되는 특성에 맞게 데이터를 파싱하고 연결하여 데이터를 전달하는 것으로 해결!



FE : 프론트엔드에서 어려웠던 점  
BE : 백엔드에서 어려웠던 점

## 팀(공통)

(BE) 사용자가 직접 시간을 설정할 때 역직렬화 문제로 LocalTime으로 전달된 데이터가 올바르게 처리되지 못했다.

입력된 데이터를 String으로 변환한 뒤, 요구하는 데이터 형태에 맞게 서버에서 다시 LocalTime 객체로 변환하여 해결!

(BE) 위의 연장선으로, 사용자가 시간 설정에 따라 가장 빠른 열차 시간이 바뀌어야 하는데, 이를 설계하는 데 어려움이 있었다.

메인에서 다른 액티비티로 넘어갈 때 입력 역 정보도 같이 전달 → 시간이 바뀔 경우 Fragment에서 서버로 바로 연결하여 변경된 정보 및 데이터를 교환하여 해결!

(BE) 수업 중 다뤘던 Spring Security를 회원 가입 및 유저 정보 저장 등으로 프로젝트에도 적용하고 싶었으나 그러지 못해서 아쉬웠다.

시간 부족의 한계임을 인정하고 다음에 더욱 상세하게 다룰 수 있도록 개인적으로 계속 공부를 할 예정!



## BSP's Development

- ✓ 프론트엔드(Front-End), 백엔드(Back-End), 데이터베이스(Database)를 모두 다룬 경험
- ✓ 공공 데이터를 활용하고 파싱한 경험
- ✓ 4,121개의 행 데이터를 파싱하여 139,102개의 행 데이터로 산출한 경험
- ✓ 문제 해결 및 개선을 위해 프로젝트 기간 동안 매일 회고록을 작성한 경험
- ✓ 팀 프로젝트로써 팀원과 커뮤니케이션하고 협업 한 경험

정에스더 | 문영민

# 부산 지 프로젝트



A photograph of a subway station platform. A train is visible through glass platform screen doors. A green circular sign on the right wall reads "금련산" (Geumjeonsan). The ceiling has a grid of lights and a red phone icon. The floor is made of large tiles.

BUSAN SUBWAY PROJECT

운행  
감사  
종료합니다.  
합니다!