|  |
| --- |
|  |
| **KNU BUS** |
|  |

모바일 앱 프로그래밍(003)

|  |  |
| --- | --- |
| 이창환 | 2016116789 |
| 신형진 | 2019114864 |
| 윤해빈 | 2018112048 |
| 허준수 | 2018115741 |

**목차**

1. KNU BUS 개요 3

2. KNU BUS 제작 배경 3

3. 기존 버스 관련 앱과의 비교 4

4. 구현 5

(1) 사용되는 언어 및 프로그램 5

(2) 프로그램 버전 정보 6

(3) 프로그램 설정 7

(4) 프로그램 액티비티 9

(5) Spring Boot 프레임워크 구현 11

(6) Android studio 코틀린 구현 21

1.KNU BUS 개요

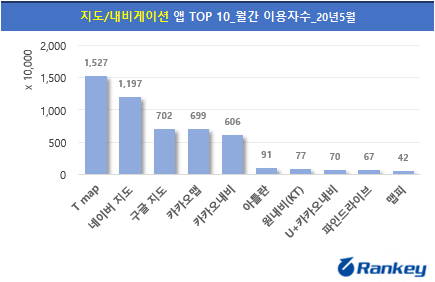
경북대학교 학생들을 위한 버스 도착 시간 알림이 앱입니다.

대구광역시의 모든 버스의 정류소별 도착 정보를 출력하여 사용자들이 정류소까지 서두르지 않고 안전하게 올 수 있도록 합니다.

2. 제작배경

기존에 존재하는 버스 관련 앱의 경우는 사용성이 그렇게 높지 않았다.

가령 버스를 타기위해 가장 가까운 정류소까지 가는데, 내가 원하는 버스가 해당 정류소까지 얼마나 남았는지를 굳이 검색해서 알아낸 다음 걷거나 뛰지는 않았다는 의미이다.



https://www.mhns.co.kr/news/articleView.html?idxno=417433

위의 앱들은 대부분의 사람들의 휴대폰에 다운받아져 있지만, 말 그대로 지도나 운전시 내비게이션의 역할에 충실하여 버스에 대한 접근성을 떨어진다고 느껴졌다.

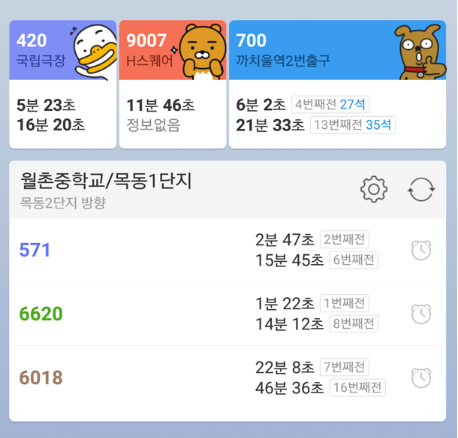
또한 정류장은 신호등 개수 및 교차로에 따라 걸리는 시간의 간격이 각 정류장들마다 상이하다.

다중 교차로를 거쳐서 오는 정류장보다 그냥 직진으로 거쳐서 오는 정류장이 빠르고, 신호등이 많은 정류장보다 적은 정류장이 대개 빠른 편이다.

이러한 이유 때문에 정류장에서 버스대기시간을 보거나 스마트폰에 앱을 통해서 정류장들을 본다면 이러한 특징들이 제대로 반영되지 않아, 도착시간의 정확도가 낮은 경우가 있다.

이를 보완하기 위해 공공데이터 포털의 버스 관련 공공 api를 사용하여 보다 편리하고 정확한 버스 도착 정보 제공 앱을 제작하겠다는 생각을 하게 되었다.

3. 기존 버스 관련 앱과의 비교



사람들이 비교적 많이 사용하는 버스 관련 앱들을 알아보자.

우선 K사의 버스 정보 제공 앱과 D사의 버스 정보 제공 앱을 가져왔다.

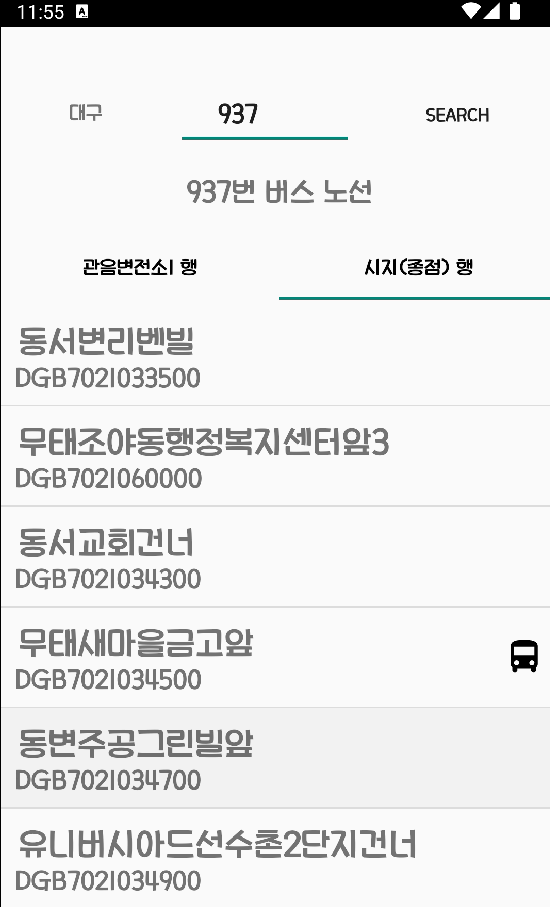
이 둘 모두 실시간 버스 위치, 주변 정류장 확인, 예상 도착 시간 등의 정보를 제공한다.

하지만 그 전국의 모든 버스를 대상으로 서비스하다 보니 매일 타는 버스만 타는 경우에는 직관적

으로 파악하기가 난감한 경우가 있다.

즉 오히려 다양한 기능과 위젯들이 사용자의 편의성을 해칠 수도 있다는 의미이다.

따라서 우리는 당초 목적에 맞는 버스 도착 예정 시간을 사용자에게 직관적으로 바로 보여줄 수 있도록 심플하지만 필수적인 앱이 되도록 구상하였다.



4. 구현

(1) 사용되는 언어 및 프로그램

프론트엔드에는 Kotlin을 사용하였으며, 백엔드에는 Java의 Spring Boot 프레임워크를 사용하였다.

텍스트, 명함, 봉투, 벡터그래픽이(가) 표시된 사진

자동 생성된 설명

그림 1 Kotlin, Spring boot

사용한 프로그램은 인텔리제이, MySQL, Android studio, Spring boot이다.

Spring 프레임워크의 어려운 XML설정과 외장 Tomcat의 단점을 보완한 Spring Boot를 사용하였다.. build.gradle의 dependencies를 이용하여 프로젝트 설정이 쉽고 자주 사용되는 라이브러리의 버전을 자동으로 관리해주며, 내장 웹서버 Tomcat을 가지고 있어 별도의 작업 없이 빠르게 서버를 실행할 수 있다. 또한 배포 시 실행가능한 jar파일로 build하여 application실행에 편리성을 제공해주기 때문에 사용하였다.

화이트보드이(가) 표시된 사진

자동 생성된 설명

데이터베이스로는 MySQL 을 사용하여, 사용자의 계정 정보를 처리하도록 하였다.

오픈소스 라이선스여서 무료일 뿐만 아니라, 다양한 embedded 된 언어를 지원해주면서 용량이 큰 데이터 집합도 효과적으로 처리할 수 있기 때문에 사용하였다.



Application 개발은 Android studio을 통해 kotlin으로 개발하였다.

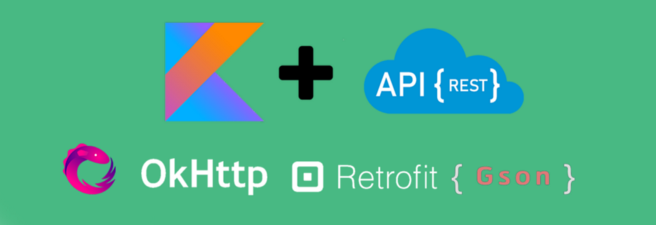
Xml을 활용하여 앱단에서 보이는 화면을 구성하기 편리하고, 모바일 앱 프로그래밍 수업에서 배운 부분을 충실하게 활용하고자 사용하였다.

(2) 프로그램 버전 정보

|  |  |
| --- | --- |
| server | Sprint Boot : 2.7.5  Java : 1.8 |
| Application | Kotlin : 213-1.7.20  Android Target SDK : 32  Android Min SDK : 28 |
| Database | MySQL : 8.0.26 |

(3) 프로그램 설정

- 외부 라이브러리



https://medium.com/android-news/manage-rest-api-with-okhttp3-retrofit2

공공 데이터 포털에서 open api를 json 타입으로 받아오는 방법을 선택했다.

따라서 이를 위해 Retrofit2(2.9.0)를 사용하였다.

Retrofit2는 REST API 통신을 위해 구현되어 HTTP 통신을 할 때 주로 사용되며, 안드로이드에서 api 서버와 통신할 때 사용되는 라이브러리이다.

또한 OkHttp(4.9.1) 또한 마찬가지의 역할을 하지만, Retrofit2 의 Parmeter, Query, Header 등의 맵핑 작업 및 반복되는 작업을 편리하게 처리함에 있어 유리하기 때문에 함께 사용하였다.

-Web

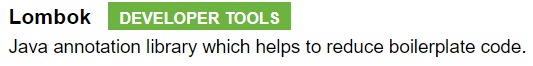
텍스트이(가) 표시된 사진

자동 생성된 설명

**Spring boot MVC패턴**

Apache Tomcat을 사용하여 내장형 웹 서버 사용

-Lombok



**Spring boot MVC패턴**

Lombok Annotation을 통한 불필요한 코드 중복 방지를 통한 가독성을 향상을 위함

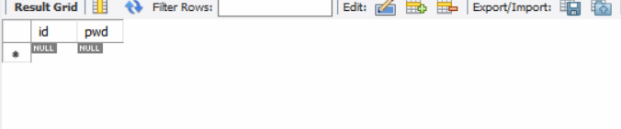
-MySQL

텍스트이(가) 표시된 사진

자동 생성된 설명

Mysql 설정

MySQL과 연동하여 데이터 저장



회원가입 전



회원가입 후

-kotlin 설정결과

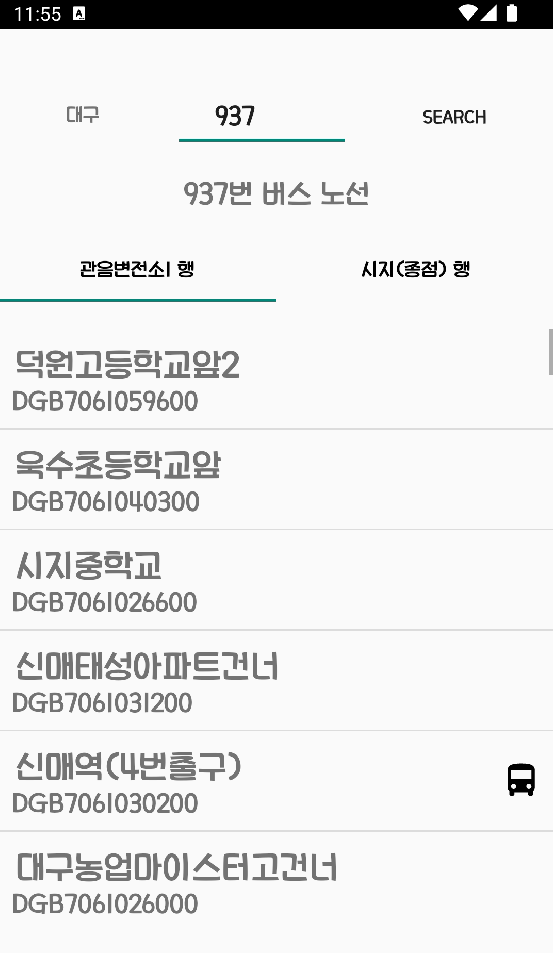
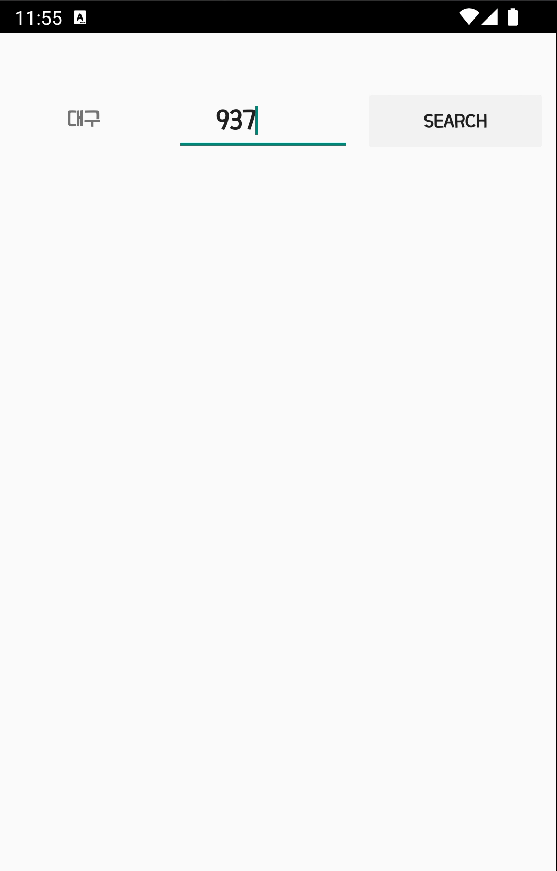
텍스트이(가) 표시된 사진

자동 생성된 설명

(4) 프로그램 액티비티



회원가입, 로그인, 로그인 시 초기화면

테이블이(가) 표시된 사진

자동 생성된 설명

노선버튼 클릭 시 노선검색 기능 및 버스 도착 정보 제공 기능

테이블이(가) 표시된 사진

자동 생성된 설명

정류소버튼 클릭 시 정류소 검색 기능 및 버스 도착 정보 제공 기능

(5) Spring Boot 프레임워크 구현

스프링부트 부분 폴더 트리구조.

텍스트이(가) 표시된 사진

자동 생성된 설명

프로그램 구성 폴더 중 핵심 폴더는 controller 와 model 폴더이다.

1. Controller

사용자가 요청한 것을 처리하고, 해당 페이지를 불러오는 코드가 정의되어 있음

2. Model

프로그램의 데이터 송수신을 위한 DTO 모델이 정의 되어있다.

핵심이 되는 API들은 아래와 같다.

**Controller**

(1) NodeController.java

(1) 정류소 아이디 반환 api

텍스트이(가) 표시된 사진

자동 생성된 설명

정류소 이름을 입력하면 해당 정류소의 id를 반환한다.

이 때 비슷한 이름인 정류소들이 다수 존재한다면 모두 출력한다.

(2) 특정 정류소 도착 예정인 모든 노선들의 정보 조회 api

텍스트이(가) 표시된 사진

자동 생성된 설명

텍스트이(가) 표시된 사진

자동 생성된 설명

특정 정류소(ex.경북대학교 정문건너)를 입력하면 해당 정류소로 진입중인 버스들의 노선번호 및 도착 예정 시간을 보여주는 코드.

(2) RouteController.java

(1) 노선이 경유하는 모든 정류소 정보 조회 api

텍스트이(가) 표시된 사진

자동 생성된 설명

텍스트이(가) 표시된 사진

자동 생성된 설명

노선의 상행, 하행에 따라 구분하여 특정 버스의 노선번호를 입력하면 해당 버스의 노선을 보여준다.

정류소 이름, 정류소 ID, 위도, 경도 정보를 반환한다.

(2) 노선별 버스 위치 목록 조회 api

텍스트이(가) 표시된 사진

자동 생성된 설명

노선별로 버스의 GPS 위치 정보의 목록을 조회한다.

버스정보와 버스가 위치하는 정류소 정보를 반환하며, 만약 존재하지 않을 시 null을 반환하도록 되어있다.

(3) 정류소별 특정 노선 버스 도착예정정보 목록 조회 api

텍스트이(가) 표시된 사진

자동 생성된 설명

정류소 별로 특정노선의 실시간 도착예정정보 및 운행정보 목록을 조회한다.

(3)loginController.java

텍스트이(가) 표시된 사진

자동 생성된 설명

유저의 회원가입 및 로그인을 정의한 코드이다.

Model

1. NodeInfo.java

텍스트이(가) 표시된 사진

자동 생성된 설명

텍스트이(가) 표시된 사진

자동 생성된 설명

정류소의 이름, ID, 해당 정류소의 위도 및 경도 그리고 정류소까지의 예상 시간을 받는 클래스를 정의하였다.

뒤에서 정의하는 route는 노선에 해당하므로 해당 클래스와는 상이함을 주의하자.

1. Nodes.java

텍스트이(가) 표시된 사진

자동 생성된 설명

상행 및 하행노선을 구분하여 클래스로 정의한 코드이다.

1. RouteInfo.java

텍스트이(가) 표시된 사진

자동 생성된 설명

노선 부분에 해당하는 코드이다.

여기에 들어가는 정류소 정보는 버스가 현재 도착예정인 가장 가까운 정류소를 의미한다.

그 외에는 진입하는 버스의 번호, id, 정류소이름 및 id, 위도 경도를 나타낼 수 있다.

1. Routes.java

텍스트이(가) 표시된 사진

자동 생성된 설명

노선 정보를 관리하기 위해 만든 클래스이다.

(6) Android studio 코틀린 구현

코틀린 부분 폴더 트리구조.

텍스트이(가) 표시된 사진

자동 생성된 설명

Spring boot 에서 작성한 REST API를 받아와서 보여주기 위한 파일들이다.

로그인 및 핵심 기능(노선검색, 정류소검색)를 모두 login 폴더에서 관리한다.

핵심 기능을 담당하는 클래스는 다음과 같다.

(1) BusRouteActivity.kt

텍스트이(가) 표시된 사진

자동 생성된 설명

Intent 를 사용하여 각 액티비티를 버튼을 클릭하면 이동할 수 있도록 하였다.

(2) BusStationActivity.kt

텍스트이(가) 표시된 사진

자동 생성된 설명

탭 레이아웃을 설정해서 상행 / 하행을 탭을 클릭할 때 이동할 수 있도록 하였다.

텍스트이(가) 표시된 사진

자동 생성된 설명

몇 번 노선인지 보여주기 위한 코드이다.

예를 들어 937 을 입력했다면 ‘937번 버스노선’ 이 가운데에 뜨며 그 아래로는 해당 노선의 노선도가 RecyclerView 에 의해서 보여지게 된다.

(3) EmgMedResponse.kt

텍스트이(가) 표시된 사진

자동 생성된 설명텍스트이(가) 표시된 사진

자동 생성된 설명

상행 하행별로 보여주는 노선도가 다르기 때문에 각각 별도의 dataclass 를 정의한 것이다.

(4) Fragement\_down & Fragement\_up.kt

텍스트이(가) 표시된 사진

자동 생성된 설명텍스트이(가) 표시된 사진

자동 생성된 설명

좌:fragment\_down 우:fragment\_up

왼쪽은 하행일 때의 노선도를 보여주기 위한 recyclerview를, 오른쪽은 상행일때의 노선도를 보여주기 위한 recyclerview를 정의한 것이다.

(5) JoinActivity.kt & Login\_User.kt & login\_api.kt

텍스트이(가) 표시된 사진

자동 생성된 설명텍스트, 스크린샷, 화면이(가) 표시된 사진

자동 생성된 설명

로그인/회원가입 화면을 보여주는 액티비티이다.

(6)NodeActivity.kt

텍스트이(가) 표시된 사진

자동 생성된 설명

REST API에서 반환받은 노선도 리스트를 json 형태로 파싱해서 그걸 화면으로 보여주는 액티비티이다.

Recyclerview를 적용한 부분.

텍스트이(가) 표시된 사진

자동 생성된 설명

도시의 경우 대구(citycode:22)로 고정해두고, route를 입력받게하여 server로 전송한다.

입력한 정류소가 존재하지 않는 경우를 위해 예외처리를 해주었다.

(7) NodeTimeActivity.kt

텍스트이(가) 표시된 사진

자동 생성된 설명

정류소별 버스 예상 도착 시간을 보여주기 위한 함수이다.

마찬가지로 해당 버스가 없는 경우를 위한 예외처리를 해주었다.

(8) RetrofitApi.kt

텍스트이(가) 표시된 사진

자동 생성된 설명

Retrofit을 사용하기 위해 정의한 object 클래스이다.

(9) busService.kt

텍스트, 모니터, 화면, 스크린샷이(가) 표시된 사진

자동 생성된 설명

4가지 GET 요청이 들어오는 경우를 인터페이스로 정의한 부분이다.

@Query 중 citicode 부분은 22로, 대구로 고정되어있으며, 각 함수에 맞는 값을 받아서 처리하게 된다.