

Open API를 활용한 주차 도우미 어플리케이션

지도교수 : 이병천 교수님

정보보호학과 91416349 임의수
정보보호학과 91416210 서태건

목차

1. 주제 선정 배경

2. 구상도

3. 개발 환경 및 개발 내용

4. 개발 시스템 운영

5. 결론 및 기대 효과

주제 선정 배경

"주차난·불법 주차 막자"...화성시, 추석 연휴동안 공영주차장 무료 개방

신창균이상문 | 입력 2021.09.16 15:52 | 수정 2021.09.16 16:10 | 댓글 0

화성시가 오는 추석 연휴기간 동안 주차난과 불법주차를 대비 개방한다고 16일 밝혔다.

시에 따르면 무료 개방 대상은 오산동 990-1번지 131면, 반송번지 65면, 남양읍 남양리 2280-1번지 65면, 서신면 제부리 202면이며, 오는 18일부터 22일까지 무료로 이용 가능하다.



덕진구 공영주차장 주변 불법 주·정차 일제 단속

이방희 기자 | 승인 2014.01.15 15:53 | 댓글 0



전주시 덕진구(구청장 노학기)는 도심지역 상습적인 주차난과 교통체증 해소를 위해 조성된 일부 공영주차장 주변에 불법 주·정차 차량으로 제 구실을 못하고 있다는 자체판단에 따라 공영주차장 주변 불법 주·정차 일제 단속에 나선다.

덕진구 관내 공영주차장은 21개소 6,960면이며, 이중 유료주차장은 5개소 1천879면, 무료주차장 16개소 5,081면의 것으로 불법 주·정차 차량이 많은 경우 이를 단속할 방침이다.

공영주차장 어디 있지?

가



기자 = 인천 을왕리해수욕장에 한 곳뿐인 공영 주차장이 입구 표장 정문에서 100여m 떨어진 곳에 위치해 있어 수많은 피서객들과 유료주차장을 이용하고 있다.

협소한 도시 내에서 아무 곳이나 주차하게 되면 주차 위반 벌금 등 여러 문제에 직면하게 됨

주제 선정 배경

행안부, '보조금24' 1000여개 보조금정보 오픈 API로 공개

등록 2021-09-14 오후 12:00:00

농협은행 API 공개로 국내 '오픈뱅킹' 탄생

승인 2021.06.29 16:38



김대훈 기자

입력 2020.09.22 17:15 | 수정 2020.09.23 01:08 | 지면 A21

손병환, 국내 첫 소스코드 공개 인증용이 정부 API로 판키워

손병환 농협은행장은 디지털금융부장 시절이던 2015년 국내 금융회사 중 최초로 오픈API 서비스를 내놓기로 했다. 오픈API란 누구든 프로그램 개발에 이용할 수 있도록 공개한 프로그래밍 명령어 묶음(소스코드)이다.

은행의 오픈API를 활용하면 핀테크 앱의 기능을 대부분 구현할 수 있다. 농협은행 '디벨로퍼 홈페이지'에 들어가면 은행 금융정보를 조회할 수 있는 서비스, 은행계좌 기반 간편결제 서비스, 개인 간(P2P) 금융에 필요한 서비스, 지로공과금 납부 등의 서비스를 구현하는 프로그래밍 코드를 내려받을 수 있다. 5년여간 농협은행 오픈API를 기반으로 63개 기업의 핀테크 서비스가 나왔고, 이들 서비스를 활용해 5조원 이상이 거래됐다.

손 행장은 "당시 금융 API를 공개(오픈)하겠다고 하니, '수수료 수익이 줄어들지 않겠나' '밥줄 끊는 일' 등의 반발이 나왔다"고 했다. 설득에는 적지 않은 시일이 걸렸다. 그는 "금융과 정보기술(IT)이 결합하면서 다양해진 소비자 요구에 맞추려면 기존 은행 서비스로는 한계가 있다"며 "농협도 구글, 애플, 알리바바처럼 국내 핀테크업계의 퍼스트무버가 될 수 있다"고 밀어붙였다.

국세청-행안부, 공공데이터 오픈API 열었다...사업자등록정보 진위확인 실시간 가능

'코로나19'로 급증한 온라인 거래가 공공데이터 기반의 사업자등록정보 진위확인 및 상태조회 오픈API 서비스를 통해 더욱 안전하고 편리해진다. 이에 따라 앞으로는 부동산 중개플랫폼에서의 허위사업자등록 여부를 쉽게 파악하는 등 온라인 상의 진위확인이 더욱 안전해질 전망이다.

29일 행정안전부(장관 전해철)와 국세청(청장 김대지)은 상거래의 근간이 되는 사업자등록정보의 진위확인 및 상태조회 서비스를 오픈API 방식으로 제공한다고 밝혔다.

오픈API(Open Application Programming Interface)란, 데이터를 사전에 정해진 형태로 제공하는 소프트웨어로, 사용자가 언제든지 필요한 형태로 가공하여 사용할 수 있도록 지원하는 데이터 제공 방법이다.

사업자등록정보의 진위여부 및 사업자의 상태정보는 경제활동을 할 때 필수적인 정보로서 그간 민간에서 수요가 매우 높았으며, 디지털 뉴딜을 계기로 행정안전부와 국세청이 협업해 적극행정으로 추진됐다.

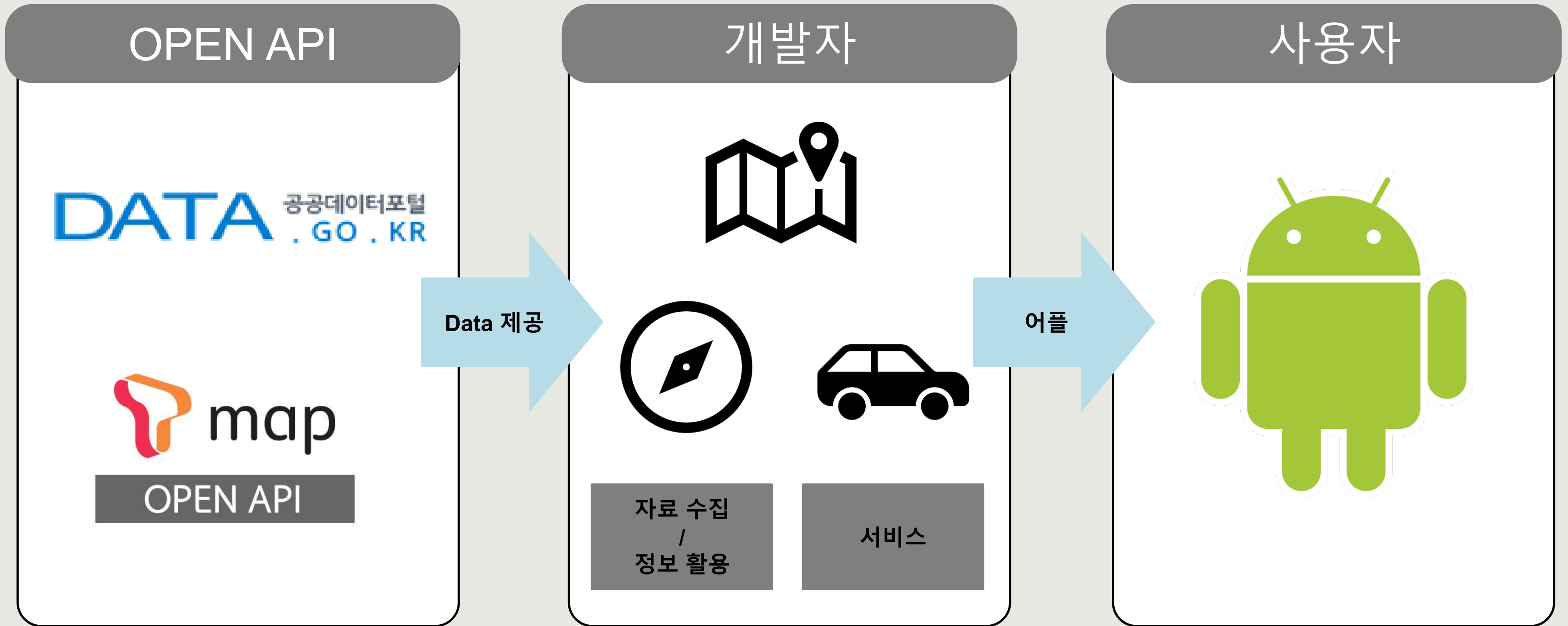
서비스는 크게 사업자등록정보 진위확인 오픈API 서비스와 사업자등록 상태조회 오픈API 서비스 등 2가지로 구성되며, 국세청이 공공데이터를 오픈API로 제공하는 최초의 서비스이다.

**오픈API를 활용한 다양한 서비스가 기업 및 정부차원에서 이루어지고 있음
그에 발 맞추어 오픈 API를 활용한 주차 도우미 어플리케이션을 구축 예정**

구상도(1/2)



구상도(2/2)



개발 환경 및 개발 내용



개발 환경 및 개발 내용

API 키 발급

Project Key (App Key)	l7xx489ee78c7b4145d2999f12770ac7ff8c
Secret Key	618d518dd4ff42b39a20eb0cf5114587
IPS	Any IP allowed

데이터포맷	JSON+XML
End Point	http://api.data.go.kr/openapi/tn_pubr_prkplce_info_api
API 환경 또는 API 호출 조건에 따라 인증키가 적용되는 방식이 다를 수 있습니다. 포털에서 제공되는 Encoding/Decoding 된 인증키를 적용하면서 구동되는 키를 사용하시기 바랍니다. * 향후 포털에서 더 명확한 정보를 제공하기 위해 노력하겠습니다.	
일반 인증키 (Encoding)	WGhgnX4FKU5%2FfGG6NOPck3W085bdIT5LDf2bEESwxweMjSK4Cup%2BePVjG2jj2%2F3jSeZUtTZD2jyvngL%2F%2FZ9Fuw%3D%3D
일반 인증키 (Decoding)	WGhgnX4FKU5/fGG6NOPck3W085bdIT5LDf2bEESwxweMjSK4Cup+ePVjG2jj2/3jSeZUtTZD2jyvngL//Z9Fuw==

T Map과 공공데이터 포털에서 API KEY발급/정보 이용 요청

개발 환경 및 개발 내용

주차장 API 연동

```
<string name="open_api_key">WGhgnX4FKU5%2FfGG6N0Pck3W085bdLT5LDf2bEESwxweMjSK4Cup%2BePVjG2jj2%2F3jSeZUtTZD2jyvnlGL%2F%2FZ9Fuw%3D%3D</string>  
<string name="tmap_api_key">l7xx489ee78c7b4145d2999f12770ac7ff8c</string>
```

```
String url = API_URL;  
url += "?serviceKey=" + getString(R.string.open_api_key); // 인증키  
//url += "&prkplceNm=" + URLEncoder.encode(this.editKeyword.getText().toString(), "UTF-8"); // 주차장명  
//url += "&prkplceSe=" + URLEncoder.encode("공영", "UTF-8"); // 주차장구분  
//url += "&prkplceType=" + URLEncoder.encode("노외", "UTF-8"); // 주차장유형  
//url += "&prkplceSe=" + URLEncoder.encode("무료", "UTF-8"); // 요금정보  
// json 타입은 주차장 검색이 안됨, (json 타입은 구조가 약간 다른 item 이 항상 배열임)  
//url += "&type=" + "json"; // xml / json  
url += "&pageNo=" + this.page; // 페이지  
url += "&numOfRows=" + ITEM_PAGE_SIZE; // 요청 데이터 수  
  
Log.d(TAG, "url:" + url);  
  
Request request = new Request.Builder()  
    .url(url)  
    .build();
```

공공 데이터포털의 전국 주차장 API를 연동하여 원하는 데이터 수집

개발 환경 및 개발 내용

지도 화면 및 위치 값 설정

```
<string name="open_api_key">WGhgnX4FKU5%2FfGG6NOPck3W085bdLT5Ldf2bEESwxweMjSK4Cup%2BePVjG2jj2%2F3jSeZUtTZD2jyvnlGL%2F%2FZ9Fuw%3D%3D</string>  
<string name="tmap_api_key">l7xx489ee78c7b4145d2999f12770ac7ff8c</string>
```

```
LinearLayout layMap = findViewById(R.id.layMap);  
// T Map  
this.tMapView = new TMapView(this);  
// T Map key 값 설정  
this.tMapView.setSKTMapApiKey(getString(R.string.tmap_api_key));  
layMap.addView(this.tMapView);  
  
this.locationManager = (LocationManager) getSystemService(Context.LOCATION_SERVICE);  
// 위치정보 사용여부 체크  
if (checkLocationServicesStatus()) {  
    // Location 초기화  
    initLocation();  
} else {  
    // 위치정보 설정값으로 보여주기  
    showLocationSettings();  
}
```

발급 받은 Key 값으로 TMapView 객체를 생성하여
지도 화면 및 위치값을 설정

개발 환경 및 개발 내용

맵 & 위치 관련

```
@Override
public void onLocationChanged(Location location) {
    Log.d(TAG, "GPS: " + "위도 " + location.getLatitude() + ", 경도 " + location.getLongitude());
    this.location = location;

    // 중심점 이동 (애니메이션 효과)
    this.tMapView.setCenterPoint(this.location.getLongitude(), this.location.getLatitude(), true)

    // 현재마커 삭제 후 다시 생성
    this.tMapView.removeMarkerItem("my");
    createMarker(this.location.getLatitude(), this.location.getLongitude(), id: "my", name: "", ti
}
@Override
```

위치 호출

어플리케이션이 내 위치값을 갱신하기 위한 코드 작성

개발 환경 및 개발 내용

맵 & 위치 관련

```
/* 마커 생성 */
private TMapMarkerItem createMarker(double latitude, double longitude, String id, String name,
    String title, String subTitle, int iconRes) {
    // 마커
    TMapMarkerItem markerItem = new TMapMarkerItem();
    // 마커 위치
    TMapPoint mapPoint = new TMapPoint(latitude, longitude);

    markerItem.setCanShowCallout(true); // 풍선뷰 사용
    markerItem.setCalloutTitle(title); // 풍선뷰 타이틀
    markerItem.setCalloutSubTitle(subTitle); // 풍선뷰 서브 타이틀

    // 마커 아이콘
    Bitmap icon = BitmapFactory.decodeResource(getResources(), iconRes);
}
```

마커 생성

맵에 표기되는 Marker 생성 및 Marker Animation 표기 적용

개발 환경 및 개발 내용

주차장 검색 기능

```
findViewById(R.id.imgSearch).setOnClickListener(view -> {
    // 주차장 검색
    if (this.location == null) {
        return;
    }

    // 주차장 검색
    Intent intent = new Intent(this, NationalParkingLotSearchActivity.class);
    intent.putExtra("latitude", this.location.getLatitude());
    intent.putExtra("longitude", this.location.getLongitude());
    this.parkingLotActivityLauncher.launch(intent);
});

findViewById(R.id.btnBookmarkAdd).setOnClickListener(view -> {
    if (this.parkingLot == null) {
        Toast.makeText(this, R.string.msg_parking_lot_empty, Toast.LENGTH_SHORT).show();
        return;
    }
});
```

위도 경도 값에 따른 주차장 검색 기능 구현

개발 환경 및 개발 내용

주차장 데이터 관련

```
// 전체 주차장 array 추가
GlobalVariable.parkingLotItems.addAll(data.response.body.items.item);

boolean download = false;
if (GlobalVariable.parkingLotItems != null) {
    if (GlobalVariable.parkingLotItems.size() > 0) {
        download = true;
    }
}
```

배열 생성

주차장 데이터 찾기에 용이하기 위한 배열 생성

개발 환경 및 개발 내용

검색 내역 저장

```
/* 검색 내역 저장 */
private void save(ParkingLot lot) {
    SQLiteDatabase db = DBHelper.getInstance(this).getReadableDatabase();

    try {
        // 내역 저장
        String sql = "INSERT INTO " + Constants.DataBaseTableName.HISTORY + "(parkingLotNo, parkingLotName, latitude, longitude, inputTimeMillis) " +
            "VALUES(?, ?, ?, ?, ?)";
        Object[] args = { lot.parkingLotNo, lot.parkingLotName, lot.latitude, lot.longitude, System.currentTimeMillis() };
        db.execSQL(sql, args);
    } catch (SQLException ignored) {
        // 오류
        Toast.makeText(this, R.string.msg_error, Toast.LENGTH_SHORT).show();
    }
    db.close();
}
```

검색 내역은 SQLite DB에 저장된다.

개발 환경 및 개발 내용

즐거찾기

```
/* 즐거찾기에 추가 */
private void addBookmark() {
    SQLiteDatabase db = DBHelper.getInstance(this).getReadableDatabase();
    long count = 0;

    // select 쿼리문 (즐거찾기에 있는지 체크)
    String sql = "SELECT count(*) FROM " + Constants.DataBaseTableName.BOOKMARK + " WHERE parking
    String[] args = { this.parkingLot.parkingLotNo };
    Cursor cursor = db.rawQuery(sql, args);
    if (cursor.moveToFirst()) {
        count = cursor.getLong(0);
    }
    cursor.close();
}
```

SQLite를 이용하여 즐거찾기를 구현

개발 환경 및 개발 내용

즐거찾기

```
if (count == 0) {
    // 즐겨찾기에 추가
    try {
        // 즐겨찾기 저장
        sql = "INSERT INTO " + Constants.DataBaseTableName.BOOKMARK + "(parkingLotNo, parkingLotName,
            "VALUES(?, ?, ?, ?)";
        Object[] args1 = { this.parkingLot.parkingLotNo, this.parkingLot.parkingLotName, this.parkingL
        db.execSQL(sql, args1);

        Toast.makeText(this, R.string.msg_bookmark_add, Toast.LENGTH_SHORT).show();
    } catch (SQLException ignored) {
        // 오류
        Toast.makeText(this, R.string.msg_error, Toast.LENGTH_SHORT).show();
    }
} else {
    Toast.makeText(this, R.string.msg_bookmark_exist, Toast.LENGTH_SHORT).show();
}

db.close();
}
```

SQLite를 이용하여 즐겨찾기를 구현

개발 환경 및 개발 내용

주차장 레이아웃

```
<LinearLayout
    android:orientation="vertical"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="140dp"
    android:layout_marginBottom="150dp"
    android:layout_gravity="bottom"
    android:gravity="center_horizontal">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textSize="15sp"
        android:text="주차장 1"
        android:textStyle="bold"
    />
```

```
<LinearLayout
    android:orientation="vertical"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="140dp"
    android:layout_marginStart="80dp"
    android:gravity="center_horizontal">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textSize="15sp"
        android:text="주차장 3"
        android:textStyle="bold"
    />

    <LinearLayout
        android:orientation="vertical"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="8dp"
        android:padding="4dp"
        android:background="@drawable/layout_parking_lot_border">

        <FrameLayout
            android:id="@+id/layParking3"
            android:layout_width="40dp"
            android:layout_height="22dp"
```

학교 현황에 유사하게 주차장과 주차 칸 배치

개발 환경 및 개발 내용

주차장 레이아웃

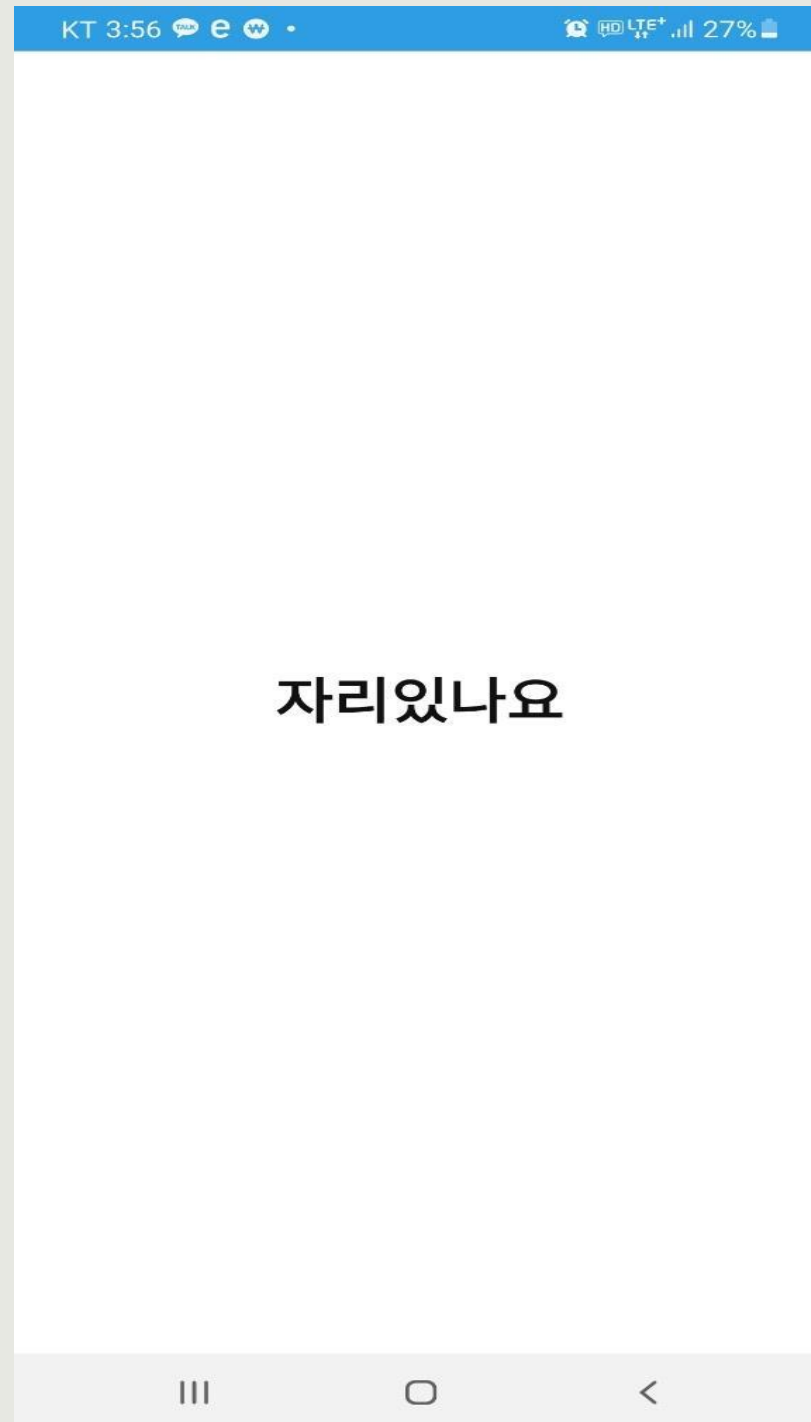
```
// QR 코드 인식 성공
String code = result.getContents();

switch (code) {
    case "A001":
        if (GlobalVariable.parkingTimeMillis1 == 0) {
            // 입차
            this.layParking1.setBackgroundResource(R.color.red_icon_color);
            this.imgParking1.setVisibility(View.VISIBLE);
        } else {
            // 출차
            this.layParking1.setBackgroundResource(R.color.blue_icon_color);
            this.imgParking1.setVisibility(View.GONE);
        }
        break;
    case "A002":
        // 주차공간 없음
        break;
    case "A003":
        if (GlobalVariable.parkingTimeMillis3 == 0) {
            // 입차
            this.layParking3.setBackgroundResource(R.color.red_icon_color);
            this.imgParking3.setVisibility(View.VISIBLE);
        } else {
            // 출차
            this.layParking3.setBackgroundResource(R.color.blue_icon_color);
            this.imgParking3.setVisibility(View.GONE);
        }
}
```

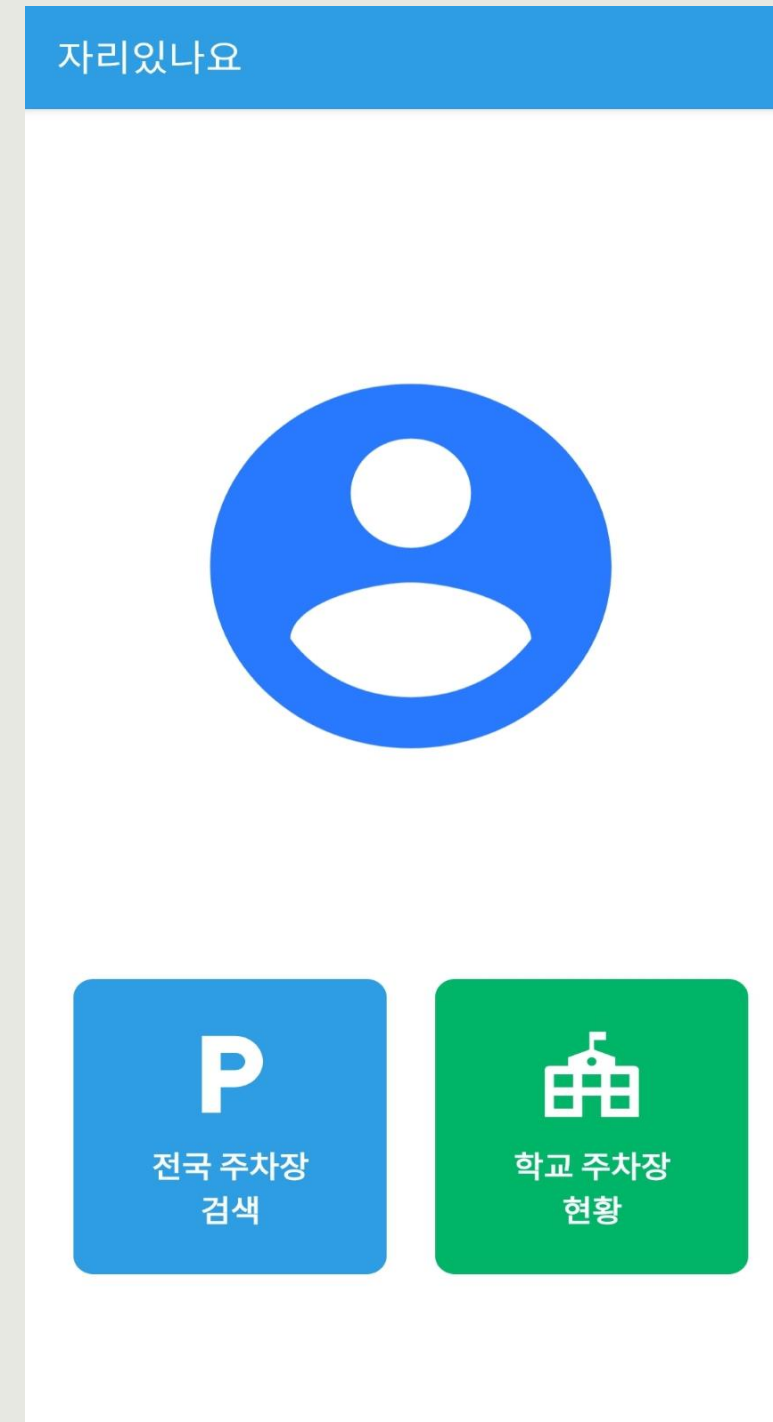
QR코드 인식을 통한 주차와 출차 현황

개발 시스템 운영

초기 화면



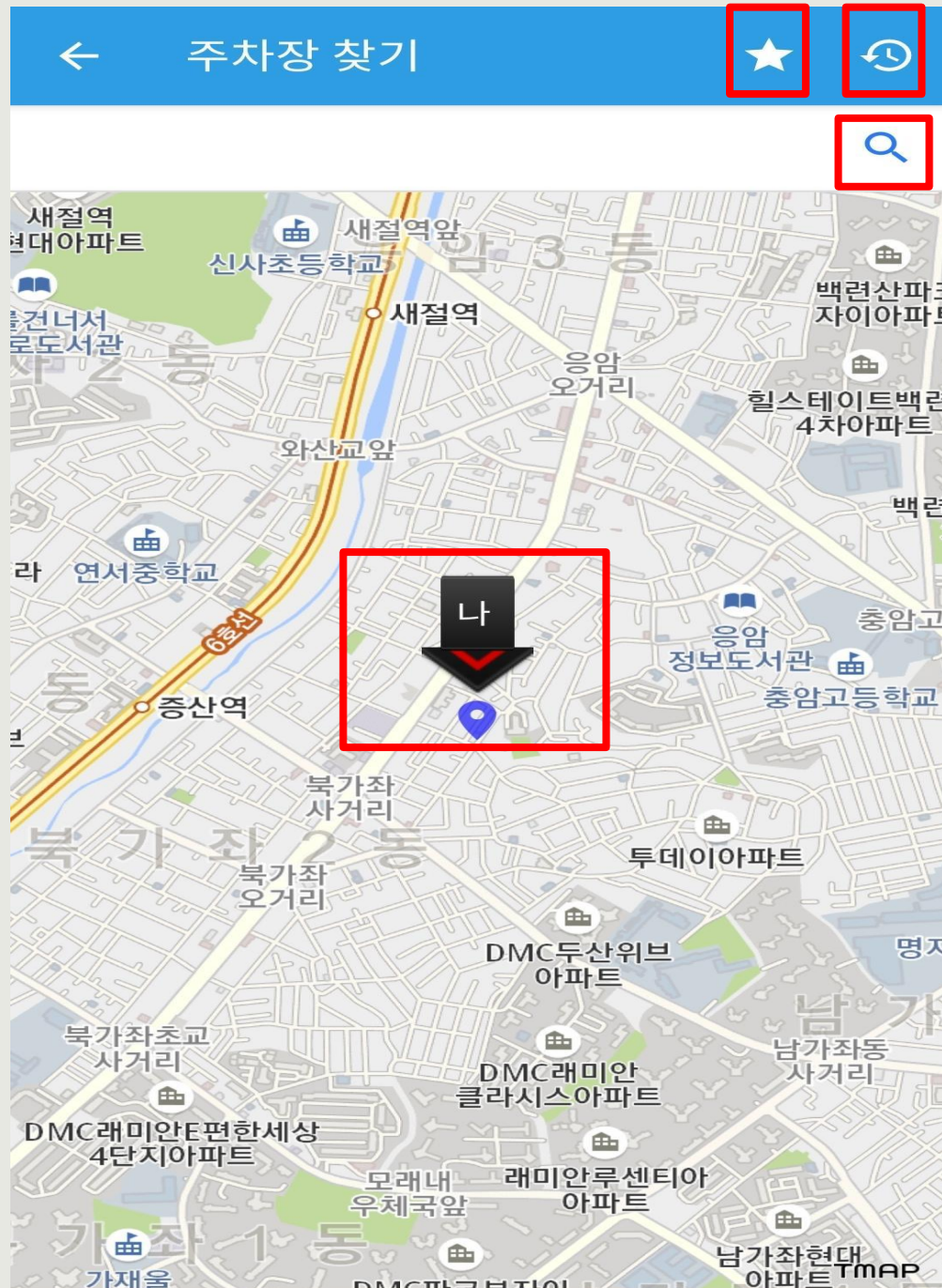
초기 화면
(어플리케이션명
자리있나요[가제목])



전국 주차장 검색
및
학교 주차장 현황
구현

개발 시스템 운영

전국 주차장 찾기



뒤로가기 기능, 즐겨찾기 기능,
최근 검색한 내역 기능 구현

돋보기 버튼을 눌러 원하는 지역 검색 가능
(동, 읍 단위 구현)

맵에 내 현재 위치 및 주차장 위치가 Marker로 표기됨

개발 시스템 운영

검색 기능 및 주차장 선택

← 주차장 검색	
서울	검색
증가로 서울특별시 서대문구 증가로 261 공영 노상 유료	578m
진흥로 서울특별시 은평구 역촌동41-8 공영 노상 혼합	2.0km
청계8가 서울특별시 중구 흥인동 167 공영 노상 혼합	5.1km
적선동 노외 서울특별시 종로구 적선동 140 공영 노외 유료	5.1km
신문로 서울특별시 종로구 새문안로 79 공영 노외 유료	5.1km
이화여고앞 공영주차장 서울특별시 중구 통일로4길 공영 노상 유료	5.1km

← 주차장 검색		
주차장구분	주차장유형	요금정보
주차장구분	주차장유형	무료
공영	노외	유료
민영	노상	혼합
	부설	

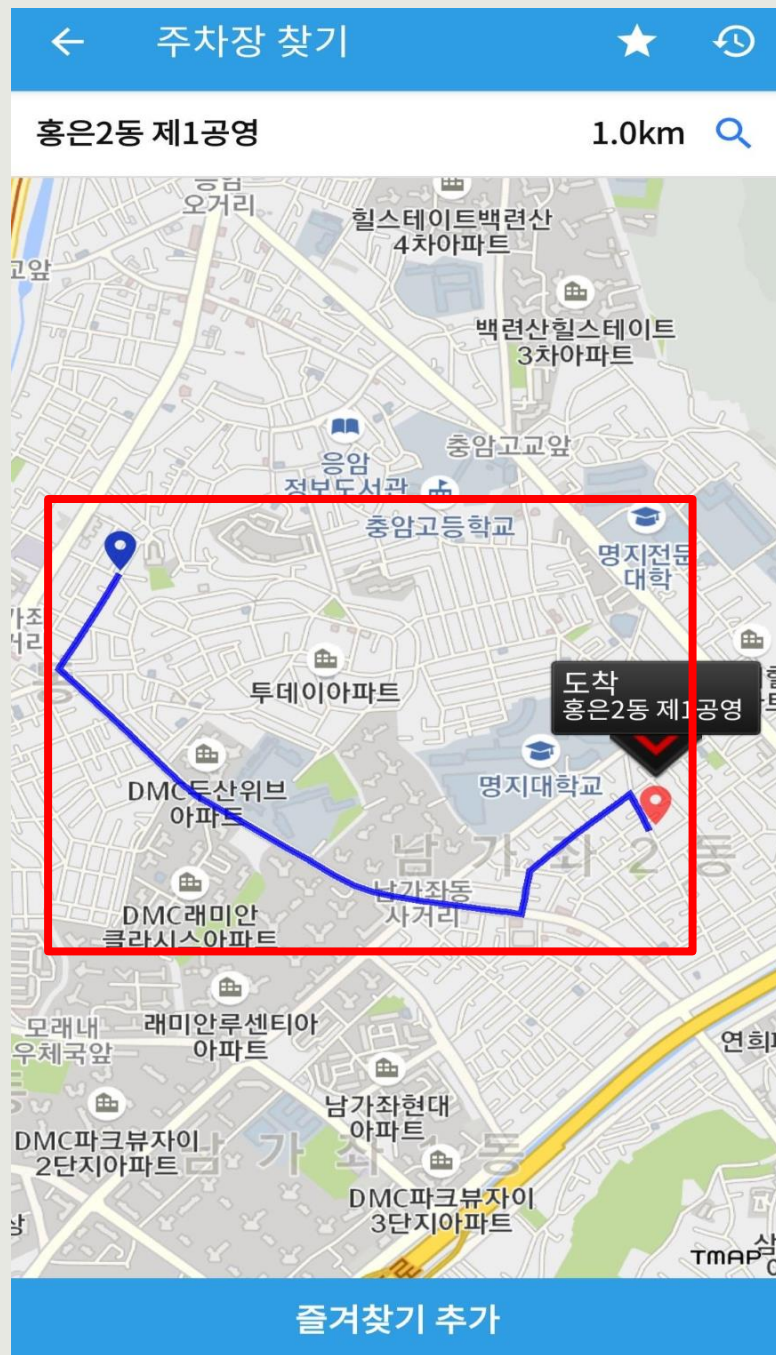
지역(도시,구,동,읍)
및 주차장 이름 검색 가능

주차장의 주소 및 종류,특징이 한 눈에 볼 수 있게
인터페이스 구현
주차장의 종류 : 노상/노외,공영,사설,혼합 등

내 현재 위치에서 주차장까지의 거리가 표기됨

개발 시스템 운영

주차장 선택 이후



내 현재 위치로부터 주차장까지의 거리가 표시되며
목적지까지 최적의 경로로 안내가 시작 됨

개발 시스템 운영

즐거찾기 등록



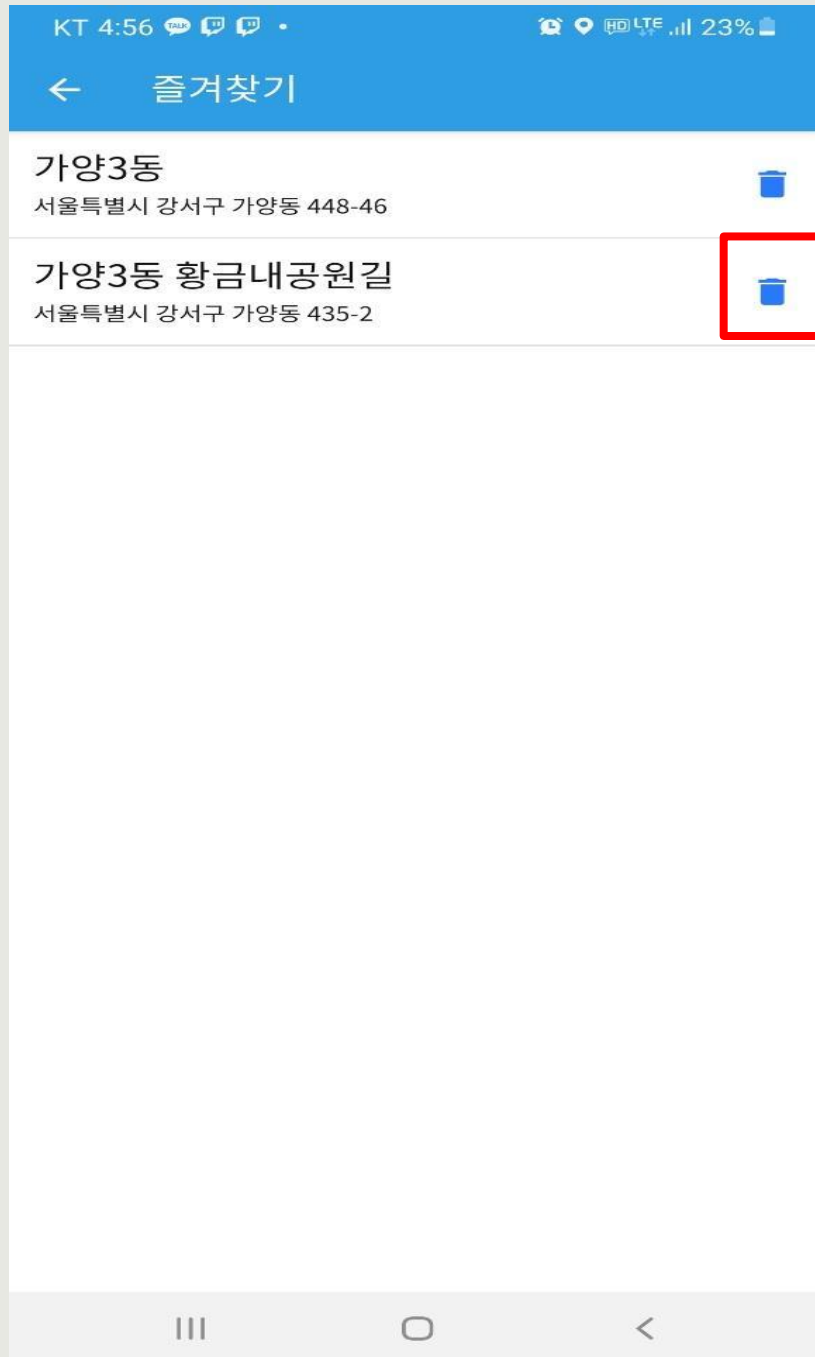
즐거찾기 버튼 클릭

즐거찾기가 추가 되었다는

텍스트박스 구현

개발 시스템 운영

즐거찾기 페이지

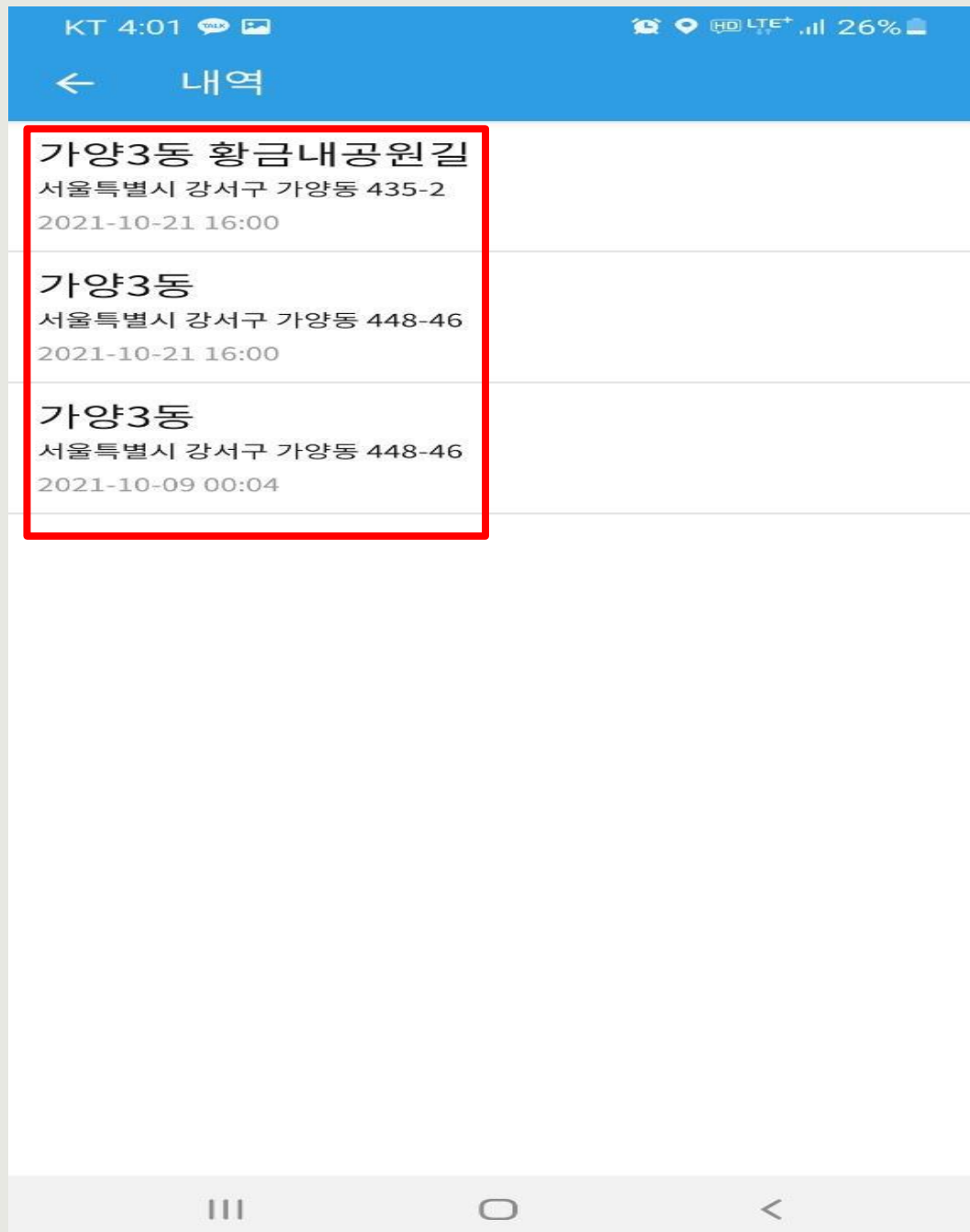


주차장 즐겨찾기 목록이 나와있는 페이지

휴지통 아이콘을 클릭하여 삭제 가능

개발 시스템 운영

최근 내역



즐거찾기 옆의 시계모양 버튼을 눌러서,

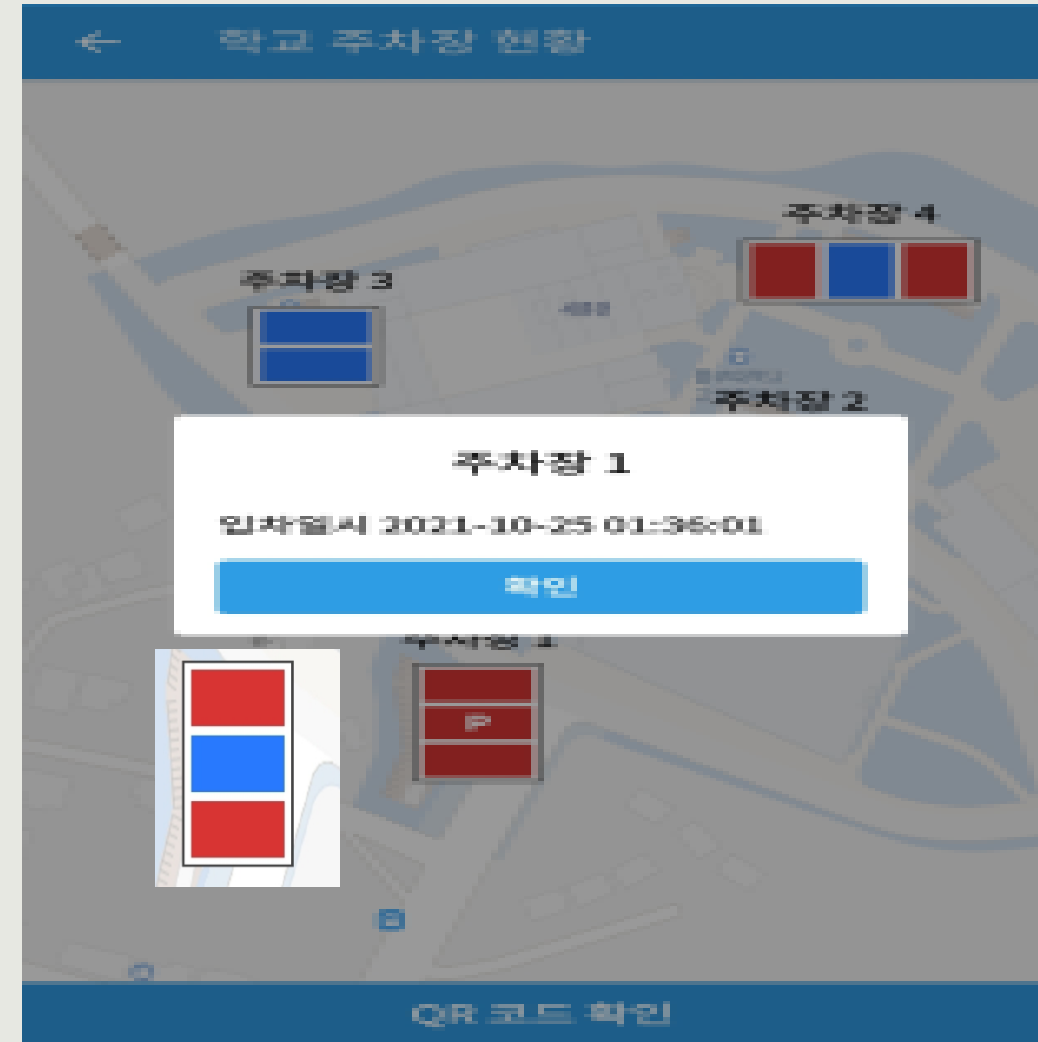
최근에 안내 받았던 주차장들의 내역을 볼 수 있음

학교 주차장 현황



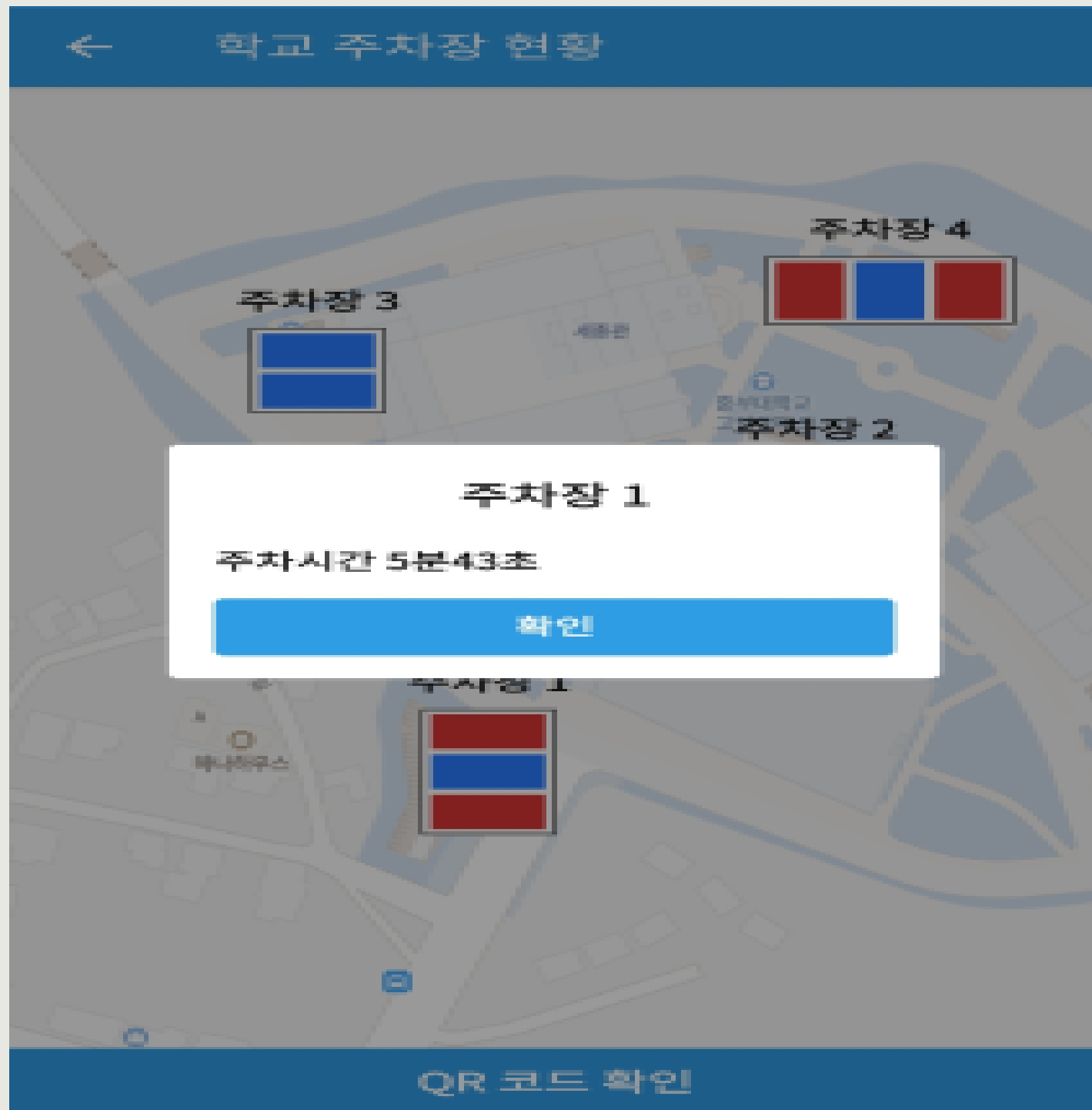
여러 개의 주차장과 주차 칸을 나누어
학교 주차장 현황을 알 수 있다.

QR 인식 (주차)



주차 마크 표시와 주차 시간 표시

QR 인식 (출차)



주차 가능 표시와 주차시간 안내

결론 및 기대효과

가구당 보유 차량은 늘어나는 반면 주차할 곳은 적거나 없는 ‘주차난’ 문제로 이웃 갈등을 겪는 사람들이 늘고 있다. 특히 아파트 주택가에서는 이중주차나 차 문을 열다가 옆 차에 흠집을 내는 이른바 ‘문콕’ 등 각종 사고까지 일어나고 있다.

최근 정부에서도 유희주차장을 개방주차장으로 운영 하는 등 다양한 정책을 내놓고 있지만, 실제 운전자들이 이러한 최근소식을 접하기 어려운게 실정이다.

본 프로젝트의 주차 어플리케이션은 일반적인 맵 어플과 차별성을 두어 주차장의 정보만을 집중적으로 가시화 하였고, 공공API를 통해 최신으로 업데이트된 주차장 정보를 받는 것에 용이하게 설계하였다. 또한 즐겨찾기 및 최근 내역을 통해 사용자의 편의성도 고려하였다.

본 프로젝트의 주차장 앱을 통해 새롭게 업데이트된 주차장을 찾고, 주차장의 정보를 한 눈에 확인하고 주차비를 비교할 수 있는데, 사용자들의 주차공간 해결과 주차에 따른 다양한 문제에 도움이 될 것이라고 기대한다.