Open API를 활용한
주차 도우미 어플리케이션

팀　　명 :　　다운비트
지도 교수 :　이병천 교수님
팀　　장 :　　임의수
팀　　원 :　　서태건

2020. 11.
중부대학교 정보보호학과

목 차

4. 결론

5. 별첨

1. 서론

1.1 연구 배경

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 유휴주차장을 개방주차장으로 운영 하는 등 다양한 정책을 내놓고 있지만, 실제 운전자들이 이러한 최근 소식을 접하기 어려운게 실정이다.
한 리서치업체의 설문 결과에 따르면, '주차앱'에 대한 고객들의 수요는 지속적으로 커지고 있다고 한다. 결과에 따르면 90% 이상의 응답자가 '주차앱'에 대해서 필요하다고 응답했다. (필요 25.2% / 매우 필요 66.5%) 이를 통해 '주차앱' 의 수요가 증가한다고 예측할 수 있었다.

이러한 주차앱의 수요에 발 맞추어 주차 도우미 어플리케이션을 만들면 어떨까 하는 생각에 연구를 시작하게 되었다.

또한 많은 국가기관과 네이버, 카카오, 구글, 페이스북을 포함한 대기업들은 지도, 음악, 날씨, 쇼핑, SNS 등 다양한 데이터를 Open API 방식으로 제공하고 있다.
이러한 Open API와 QR Code를 응용한다면 최신 업데이트 된 주차장의 정보를 사용자들에게 빠르게 제공할 수 있을 것이라 생각이 되었다. 그리하여 Open API를 활용한 주차 도우미 어플리케이션에 관한 프로젝트를 시작하게 되었다.

1.2 연구의 필요성

운전을 하여 여행을 가거나 처음 가보는 도시를 가게 된다면 주차할 장소를 찾지 못해 난감한 적이 있었다. 그럴 때 자신의 주변에 있는 가까운 주차장을 알려주는 애플리케이션이 있으면 주차장을 찾는 수고가 덜 수 있을 것이라고 예상하였다.
또한 우리 학교 주차장에서도 자리가 없을 때 옆 주차장의 주차 현황을 알려 준다면 학생들이 편하게 이용할 수 있을거라고 예상되었다.
그러한 주차 공간과 주차문제 해결에 특화된 전용 어플리케이션이 필요하다고 생각하였다.

1.3 연구 목적

'공공데이터포털'에서 제공하는 '전국주차장정보표준데이터' API와 SK에서 제공하는 T Map API 두 가지의 Open API를 이용하여 전국 주차장 검색 기능을 만드려고 한다.
GPS 정보를 활용하여 사용자의 현재 위치에 따른 주차장까지의 거리를 표기하고, 주차장의 정보를 한 눈에 볼 수 있는 유저 친화적인 주차장 어플리케이션을 만드려 한다.
세부적인 목표로는 맵에 현재 위치를 표기할 수 있는 marker를 넣고, 맵의 크기를 확대하고 축소 할 수 있는 기능을 구현한다. 즐겨찾기, 검색내역 등 유저 친화적인 UI를 구성하려한다. 또한 QR코드를 활용하여 학교 주차장의 현황을 알 수 있게 한다.

1.4 성과 요약

안드로이드 스튜디오로 Open API와 QR코드 기능을 이용하여 주차도우미 어플리케이션을 만들었는데, 이 어플리케이션을 통해 주차장 정보를 한 눈에 확인하고 주차장을 비교할 수 있는 UI를 만들었다.

자신의 현재위치에 따른 주차장까지의 최적 경로를 알 수 있으며, 맵을 확대/축소 할 수 있는 편의성을 갖추었다.

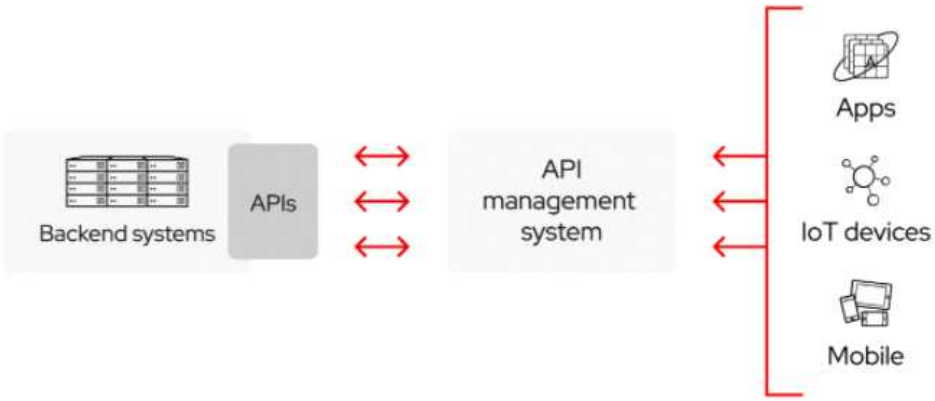
또한 즐겨찾기 기능과 최근 검색한 내역 등을 확인할 수 있는 유저 친화적인 기능을 구현하였다.

QR코드를 통해 자신이 학교에서 주차한 자리를 확인하고, 총 주차한 시간을 확인할 수 있는 기술을 구현하였다.

2. 관련 연구

2.1 Open API

Open API는 누구나 사용할 수 있도록 공개된 API를 말한다. API는 (Application Programming Interface)의 줄인 말로 어떠한 응용 프로그램에서 데이터를 주고받기 위한 방법을 의미한다. 오픈 API와 비공개 API로 나누어지며 특정 서비스를 이용할 때 데이터를 공유할 경우 어떠한 방식으로 정보를 요청하고 받을 수 있는지에 대한 규격들을 API라고 부른다.



데이터를 제공하는 주체는 제공할 모든 데이터들의 정보를 한곳으로 통합하여 수집하며 각자의 정한 절차를 통해 데이터를 관리하고 데이터 리소스에 대한 액세스 권한을 제공하고 보안과 제어를 유지할 수 있게 해준다. 액세스 권한을 어떻게, 누구에게 제공할지에 대한 여부는 API Key 발급을 통해 애플리케이션 또는 사용자를 식별하는 고유의 코드를 발급하여 API의 사용을 추적 및 제어할 수 있도록 하여 악의적 사용이나 남용을 방지한다.

2.2 QR코드

QR 코드는 컴퓨터가 만든 흑백 격자무늬 패턴 코드로, 정보를 나타내는 매트릭스 형식의 이차원 코드이다. 비슷한 용도로 먼저 사용된 이 차원 코드로는 바코드가 있다. 바코드는 이름 그대로 단순한 막대기 모양의 바를 이차원으로 나열한 것이다. QR코드는 주로 한국, 일본, 중국, 영국, 미국 등에서 많이 사용되며 명칭은 덴소 웨이브의 등록 상표 'Quick Response'에서 유래하였다. 종래에 많이 쓰이던 바코드의 용량 제한을 극복하고 그 형식과 내용을 확장한 2차원의 패턴으로 종횡의 정보를 가져서 숫자 외에 문자의 데이터를 저장할 수 있다.

QR코드 모델1 모델2	Micro QR코드	iQR코드	SQRC	Frame QR
				
【특징】 모델1은 가장 처음 만들어진 QR코드. 최대 버전은 14(73 X 73셀)로 1167자리의 숫자까지 취급이 가능합니다. 모델2는 모델1을 개량하여 만든 것으로, 최대 버전은 40(177 X 177셀)이고 7089자리의 숫자까지 취급할 수 있는 코드입니다. 현재 QR코드라 하면 일반적으로 모델2를 가리킵니다.	【특징】 위치 찾기 심벌이 하나이며, 보다 더 작은 공간에 인쇄를 가능하게 한 QR코드입니다. 마진(여백)도 2셀분만 있으면 충분히 기능을 발휘합니다.(QR코드는 최소 4셀분의 마진이 코드 주위에 필요). 최대 버전은 M4(17 X 17셀)로 35자리의 숫자까지 취급할 수 있습니다.	【특징】 정방형/장방형 생성이 가능하며, 표리반전/흑백 반전/도트 패턴(직접 부품 마크)으로 인쇄가 가능합니다. 이론상의 최대 버전은 61(422 X 422셀)로 약 4만 자리의 숫자까지 취급할 수 있습니다.	【특징】 데이터 인식 제한 기능을 가진 코드입니다. 개인 정보나 사내 정보 관리 등에 활용할 수 있습니다. 겉모양은 보통 QR코드와 같습니다.	【특징】 코드 안에 자유롭게 사용할 수 있는 캔버스 영역을 가진 QR코드입니다. 캔버스 부분에 문자나 화상을 넣을 수 있어, 판매 촉진용 돌이나 진위 판정 코드 등, 여러 가지 용도로 이용할 수 있습니다.

QR코드의 구조 일반 바코드는 단방향 즉, 1차원으로 숫자나 문자 정보가 저장 가능한데 QR코드는 종횡으로 2차원 형태를 가져서 더 많은 정보를 가질 수 있으며, 사진, 동영상, 지도, 명함 등 다양한 정보를 더 편리하게 담아낼 수 있다. 버전1부터 버전40까지 다양한 버전을 지원하고 버전마다 최대로 포함할 수 있는 정보와 크기가 다르다. QR코드에는 데이터의 표현과 읽기를 수월하게 하고자 콰이어트 존, 위치 검출 패턴 (분리자 포함), 타이밍 패턴, 정렬 패턴, 포맷 정보, 버전 정보, 데이터 영역 (에러 정정 코드 영역 포함) 등의 영역이 나뉘어 있다.

2.3 Java

썬 마이크로시스템즈에서 1995년에 개발한 객체 지향 프로그래밍 언어. 창시자는 제임스 고슬링이다. 2010년에 오라클이 썬 마이크로시스템즈을 인수하면서 Java의 저작권을 소유하였다. C#과 문법적 성향이 굉장히 비슷하며, 그에 비해 2019년 Q3에서 가장 많이 이용하는 언어로 뽑혔다.

2.4 XML

XML(Extensible Markup Language)은 W3C에서 개발된, 다른 특수한 목적을 갖는 마크업 언어를 만드는데 사용하도록 권장하는 다목적 마크업 언어이다. XML은 SGML의 단순화된 부분집합으로, 다른 많은 종류의 데이터를 기술하는 데 사용할 수 있다. XML은 주로 다른 종류의 시스템, 특히 인터넷에 연결된 시스템끼리 데이터를 쉽게 주고받을 수 있게 하여 HTML의 한

계를 극복할 목적으로 만들어졌다. W3C는 XML 설계 목표에서 단순성과 일반성, 그리고 인터넷을 통한 사용가능성을 강조했다. XML은 텍스트 데이터 형식으로 유니코드를 사용해 전 세계 언어를 지원한다.

2.5 JSON

JSON(JavaScript Object Notation)은 속성-값 쌍(attribute–value pairs and array data types (or any other serializable value)) 또는 "키-값 쌍"으로 이루어진 데이터 오브젝트를 전달하기 위해 인간이 읽을 수 있는 텍스트를 사용하는 개방형 표준 포맷이다. 비동기 브라우저/서버 통신 (AJAX)을 위해, 넓게는 XML(AJAX가 사용)을 대체하는 주요 데이터 포맷이다. 특히, 인터넷에서 자료를 주고 받을 때 그 자료를 표현하는 방법으로 알려져 있다. 자료의 종류에 큰 제한은 없으며, 특히 컴퓨터 프로그램의 변수값을 표현하는 데 적합하다.

2.6 SQLite

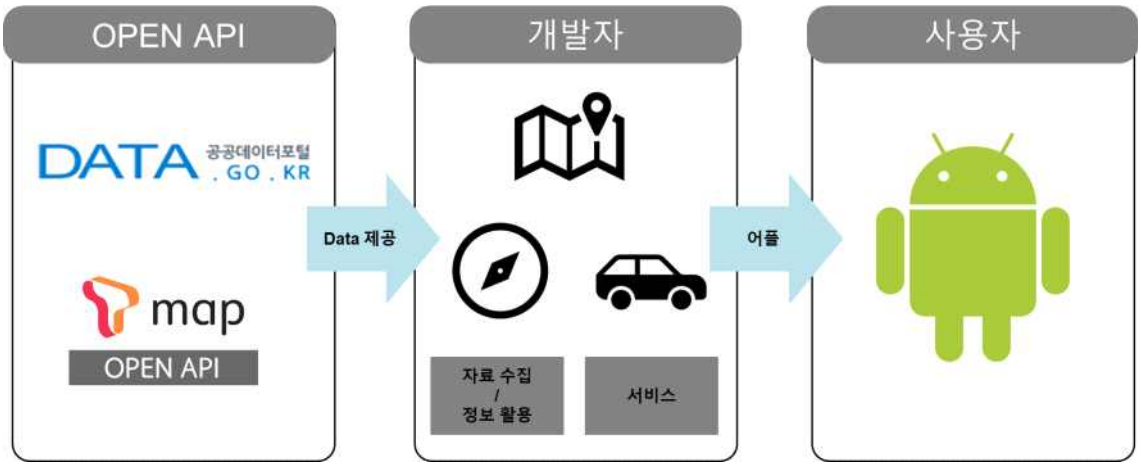
SQLite는 구글 안드로이드 운영 체제에 기본 탑재된 데이터베이스로 MySQL나 PostgreSQL와 같은 데이터베이스 관리 시스템이지만, 서버가 아니라 응용 프로그램에 넣어 사용하는 비교적 가벼운 데이터베이스이다.
일반적인 RDBMS에 비해 대규모 작업에는 적합하지 않지만, 중소 규모라면 속도에 손색이 없다. 또 API는 단순히 라이브러리를 호출하는 것만 있으며, 데이터를 저장하는 데 하나의 파일만을 사용하는 것이 특징이다.

2.6 OKhttp3

OkHttp는 HTTP 클라이언트이다. 쉽게 말하여 HTTP 기반의 request/response를 할 수 있도록 도와주는 오픈소스 라이브러리이다.
동기, 비동기 방식을 각각 제공하여 개발자가 선택하여 개발할 수 있다.
외부에 있는 데이터를 받아오기 위하여 사용하였다.

3. 본론

3.1 서비스 설계



1) 이용자에게 주차장의 위치와 정보를 알려주고 그곳으로의 지도의 목적지로 길 안내를 하는 서비스를 목적으로 하기 위한 데이터를 API 관리 주체에게 사용 승인을 얻어야 한다.

2) API의 악의적 사용을 막고 추적 및 제어를 효과적으로 할 수 있도록 주체에게 API Key를 요청하고 고유의 인증 Key를 부여받아 제공하는 데이터 이용할 수 있다.

3) 주체에게 사용을 허가받은 API 데이터 정보 중 서비스를 간략하고 신속하게 할 수 있도록 원하는 데이터 값들만 추출하여 사용자들이 이용할 수 있게 앱으로 제공한다.

4) 학교 주차장에 맞게 주차공간과 주차장의 칸을 만들어 이용자가 편리하게 식별 가능하고 QR코드를 사용하여 자신의 주차 지역과 출차 시 주차시간에 대한 정보를 알 수 있다.

5) 사용자는 앱으로 서비스를 이용하며, 원하는 목적인 사용자의 원하는 조건의 주차장의 위치와 학교 내에서의 주차장 현황을 제공하여 사용자가 원하는 서비스를 제공한다.

3.2 구현 과정 설명

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

T MAP api key 값

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 구분되는 키를 사용하시기 바랍니다. * 향후 포털에서 더 명확한 정보를 제공하기 위해 노력하겠습니다.	
일반 인증키 (Encoding)	WGhgnX4FKU5%2FfGG6NOPck3W085bdlT5LDf2bEESwxweMjSK4Cup%2BePVjG2jj2%2F3jSeZUtTZD2jyvnGL%2F%2FZ9Fuw%3D%3D
일반 인증키 (Decoding)	WGhgnX4FKU5/fGG6NOPck3W085bdlT5LDf2bEESwxweMjSK4Cup+ePVjG2jj2/3jSeZUtTZD2jyvnGL//Z9Fuw==

공공데이터 전국 주차장 현황 key 값

```
<string name="open_api_key">WGhgnX4FKU5%2FfGG6NOPck3W085bdlT5LDf2bEESwxweMjSK4Cup%2BePVjG2jj2%2F3jSeZUtTZD2jyvnGL%2F%2FZ9Fuw%3D%3D</string>
<string name="tmap_api_key">l7xx489ee78c7b4145d2999f12770ac7ff8c</string>
```

1) T map과 공공데이터에서 발급 받은 api의 키 값을 이용하여 이 프로젝트에서 사용 가능하게 한다.

```
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 다름 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();
```

2) 공공데이터의 전국 주차장 현황 api가 제공 하는 데이터 중 원하는 데이터 값을 수집한다.(XML과 JSON형식)

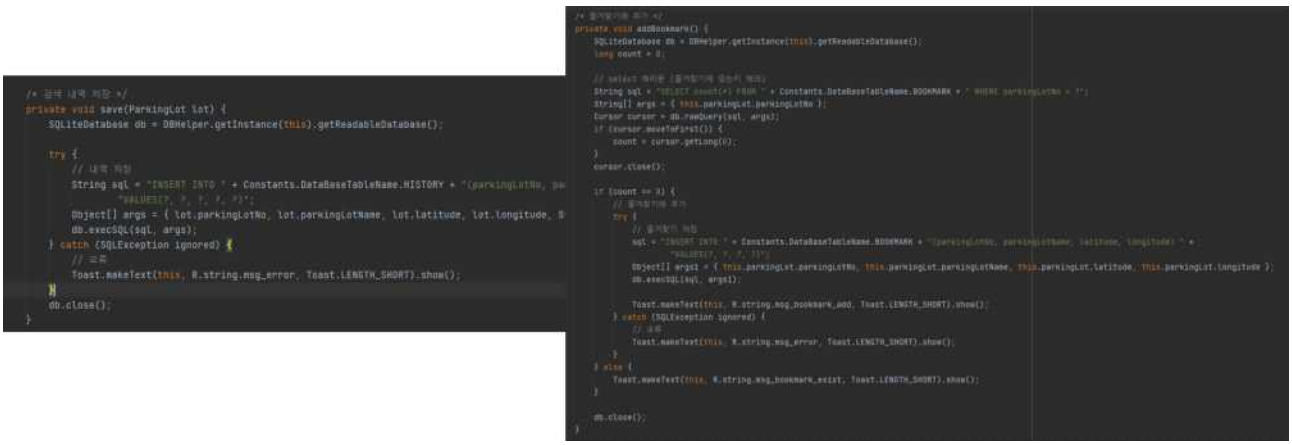
```
LinearLayout layMap = findViewById(R.id.layMap);
// T Map
this.tMapView = new TMapView(this);
// T Map key 값 설정
this.tMapView.setSKTMapApiKey(getString(R.string.tmap_api_key));
layMap.addView(this.tMapView);

this.locationManager = (LocationManager) getSystemService(Context.LOCATION_SERVICE);
// 위치정보 사용여부 체크
if (checkLocationServicesStatus()) {
    // Location 초기화
    initLocation();
} else {
    // 위치정보 설정값으로 보여주기
    showLocationSettings();
}
}
```

3) 발급 받은 key값으로 T맵을 생성하여 지도 화면 및 위치값을 설정한다.

```
        // 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;
    }
}
```

4) 15,000여건이 되는 주차장 정보를 매번 가져오면 시간이 오래 걸리므로 한번 불러온 정보는 array에 추가되어 다시 검색을 하여도 array에 저장되어 있는 정보를 불러온다.



5) 검색 내역과 즐겨찾기 기능은 임베디드 시스템인 SQLite를 이용하여 DB에 저장되어 지난 검색 내역을 확인 할 수 있고 자주가는 주차장을 등록하여 빠르게 이용 가능 하다.

```
<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"
        />
```

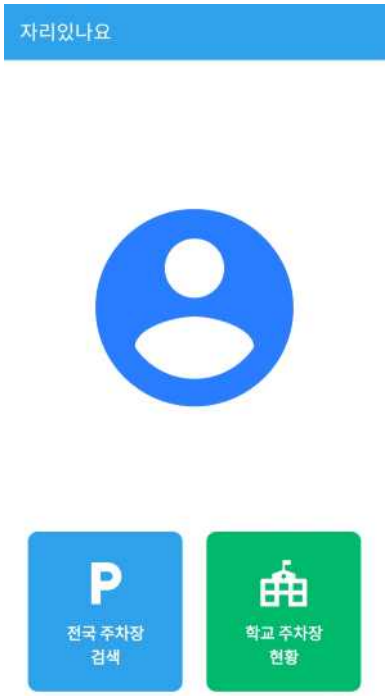
```
<LinearLayout
    android:orientation="vertical"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="140dp"
    android:layout_marginStart="80dp"
    android:gravity="center_horizontal">

    <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"
```

6) 사용자가 주차장의 현황을 식별 할 수 있도록 학교주차장과 유사한 여러 주차장과 주차 칸을 만들어 현황을 알 수 있게 하였다.

```
// 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);
        }
```

7) QR코드를 처음 인식하게 되면 해당 칸에 빨간색으로 표시되어 자리가 없음을 알리고 자신의 주차된 자리에 P 마크가 남게 된다. 다시 인식하게 되면 파란색으로 표시가 되고 총 주차시간이 나오게 된다.

3.3 주요 기능 데모



초기 화면에서 전국 주차장 검색과 학교 주차장 현황 중 원하는
기능을 선택하여 이용

3.3.1 전국 주차장 현황 기능

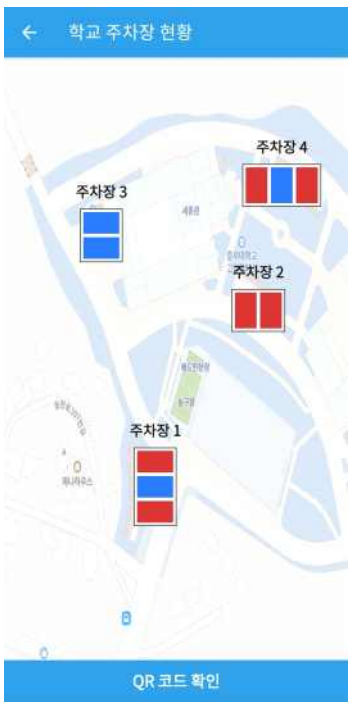
1. 사용자의 위치 정보를 수집할 수 있는 권한을 승인하면 해당 사용자의 위치가 마커로 표시되고, 휴대폰의 임베디드 시스템인 SQLite를 이용하여 즐겨찾기와 검색내역이 자체 db에 저장된다.

2. 돋보기 모양 버튼을 누르고 주차장 검색 시 필터를 사용하여 원하는 조건의 주차장을 검색할 수 있도록 함.

3. 필터에 원하는 조건을 고른 뒤 찾고 싶은 지역의 키워드를 입력하게 되면 거리 순으로 주차장이 나열되고 원하는 주차장을 고르게 되면 주차장의 정보가 나오게 된다.

4. 길 찾기를 누르게 되면 선택한 주차장까지의 효율적인 길로 안내가 시작되며 근처 원하는 주차장으로 길안내가 시작되며 자주 가는 주차장일 경우 즐겨찾기에 추가할 수 있다.

3.3.2 학교 주차장 현황 기능



1. 사용자의 카메라 사용 권한을 승인하게 되면 qr코드 인식이 가능하고 학교 주차장의 현황을

그림과 같이 볼 수 있다.

2. 자리의 qr코드를 인식하면 해당 주차장의 해당 칸에 빨간색으로 자리가 표시되고 주차한 자리에 P마크가 생기며 주차한 시간이 나오게 된다.

3. 다시 qr코드를 인식하면 출차로 인식되며 해당 칸에 파란색으로 표시가 되고 자리가 있음을 알리고 총 주차 시간이 표시된다.

4. 결론

4.1 결론 및 기대효과

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.

4.2 향후 계획

1) 학교 주차장 현황에서 현재는 해당 디바이스에서만 학교 주차장 현황을 확인할 수 있지만 향후 서버를 연결하여 다른 사용자의 디바이스에도 실시간으로 주차와 출차를 확인할 수 있도록 할 계획이다.

2) 전국 주차장 현황에서 주차장을 선택하여 먼 곳의 목적지를 안내할 때 지도의 실시간 데이터를 통해 더욱더 효율적인 길 안내를 목표로 하고 있다.

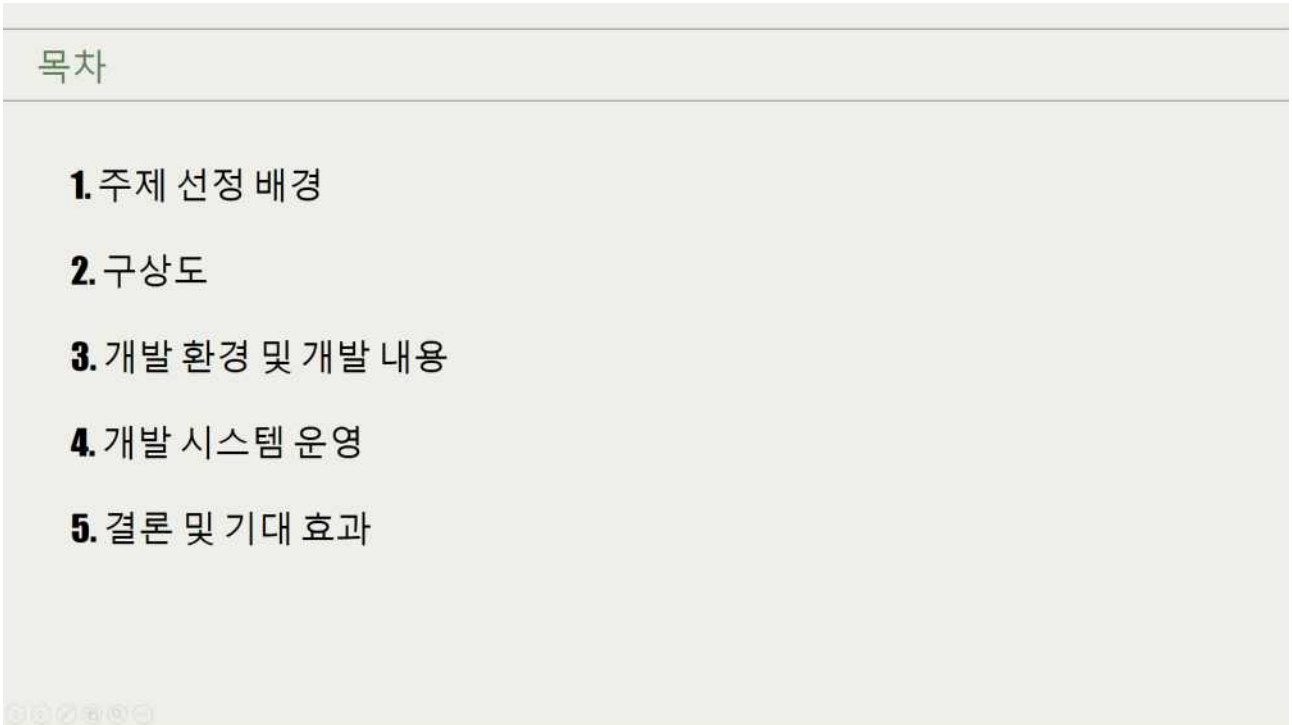
5. 별첨

5.1 소스 코드

https://github.com/tgseo76

5.2 발표 자료


공공 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

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

주제 선정 배경

행안부, '보조금24' 1000여개 보조금정보 오픈 API로 공개

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

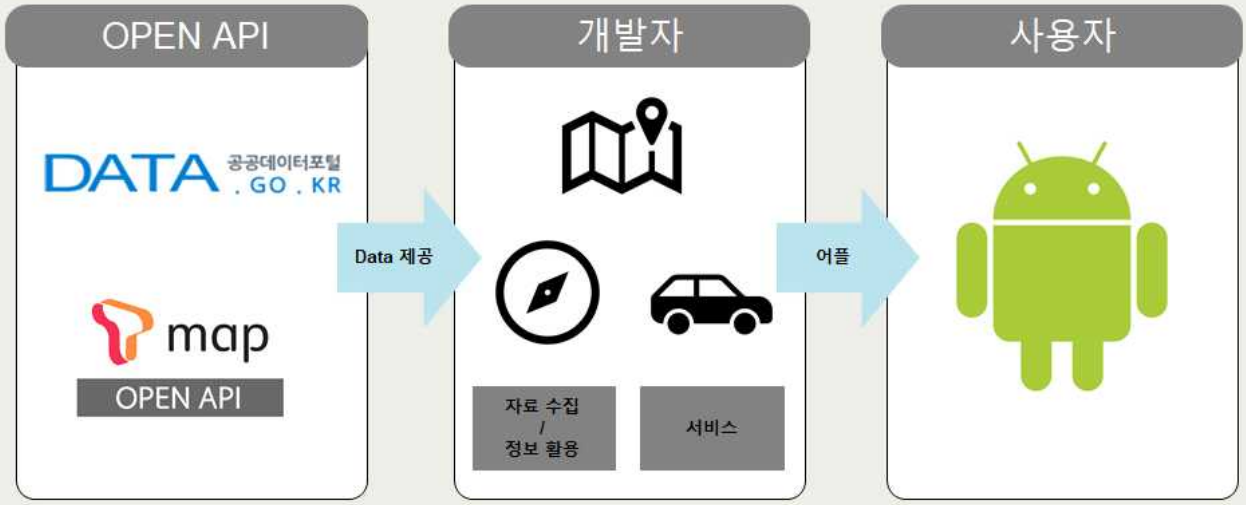
농협은행 API 공개로 국내 '오픈뱅킹' 탄생

손병환, 국내 첫 소스코드 공개
임종룡이 정부 API로 판키워

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%2FfGG6NOPck3W085bdlT5LDf2bEESwxweMjSK4Cup%2BePVjG2jj2%2F3jSeZUtTZD2jyvnGL%2F%2FZ9Fuw%3D%3D
일반 인증키 (Decoding)	WGhgnX4FKU5/fGG6NOPck3W085bdlT5LDf2bEESwxweMjSK4Cup+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%2FfGG6NOPck3W085bdlT5LDf2bEESwxweMjSK4Cup%2BePVjG2jj2%2F3jSeZUtTZD2jyvnGL%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 다름 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%2FfGGoNOPck3W085bdlT5LDf2bEESwxweMjSK4Cup%2BePVjG2jj2%2F3jSeZUtTZD2jyvnGL%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

위치 호출

```
@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: "", t
}

@Override
```

어플리케이션이 내 위치값을 갱신하기 위한 코드 작성

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

```
배열 생성

        // 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;
        }
    }
```

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

즐겨찾기

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

    // select 쿼리문 (즐겨찾기에 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 즐겨찾기를 구현

개발 환경 및 개발 내용

즐겨찾기

```
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

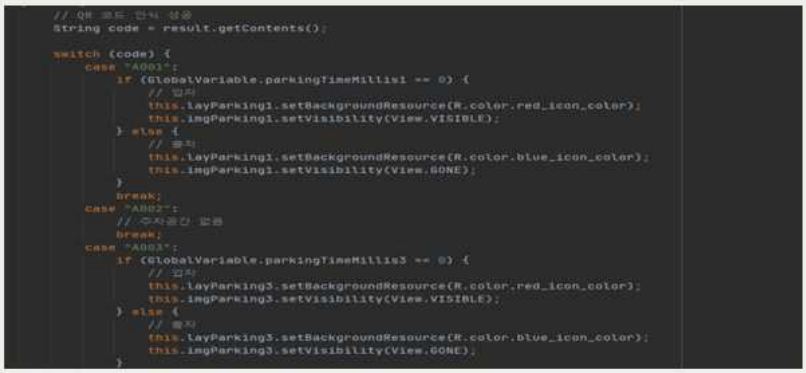
주차장 레이아웃

```
<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"
        />
```

학교 현황에 유사하게 주차장과 주자 칸 배치

개발 환경 및 개발 내용

주차장 레이아웃



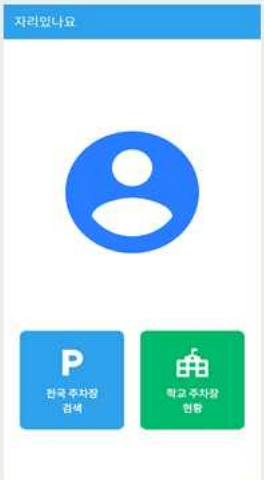
```
// QR 코드 인식 성공
String code = result.getContents();

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

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 경로로으로 안내가 시작 됨

개발 시스템 운영

즐겨찾기 등록

즐겨찾기 버튼 클릭

즐겨찾기가 추가 되었다는

텍스트박스 구현

개발 시스템 운영

즐겨찾기 페이지

주차장 즐겨찾기 목록이 나와있는 페이지

휴지통 아이콘을 클릭하여 삭제 가능

개발 시스템 운영

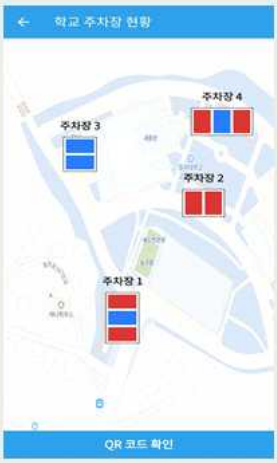
최근 내역

즐겨찾기 옆의 시계모양 버튼을 눌러서,

최근에 안내 받았던 주차장들의 내역을 볼 수 있음

개발 시스템 운영

학교 주차장 현황

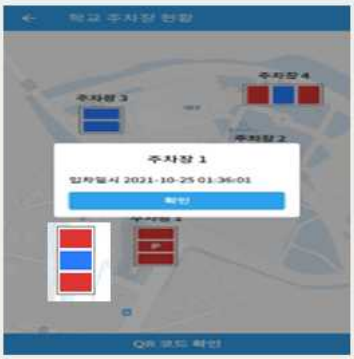


여러 개의 주차장과 주차 칸을 나누어

학교 주차장 현황을 알 수 있다.

개발 시스템 운영

QR 인식 (주차)



주차 마크 표시와 주차 시간 표시

개발 시스템 운영

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 유휴주차장을 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.