Pusan National University
Computer Science and Engineering
Technical Report 2022-31

Wi-SUN Mesh 네트워크를 이용한 주차공간 찾기 시스템 구축

201724438 김정수

201924490 서경우

201724605 최재원

지도교수 정상화 교수님

목 차

1. 서론

1.1. 연구 배경

현재 세계 각국에서 도시 경쟁력 향상을 위해 스마트시티 사업을 진행중이다. ICT기술을 접목한 최첨단 도시 모델로 교통 혼잡, 에너지 부족 등 급격한 도시 성장에 따른 각종 도시문제를 해결하고 고용 창출, 해외 수출 등으로 경제 성장 동력 역할을 할 것으로 기대된다.

[그림1]스마트시티 시장규모 예측

국내에서도 서울, 부산, 세종시 등에서 단편적인 공공서비스 수준을 넘어 민간 참여를 통한 서비스 고도화를 목표로 스마트 시티에 대한 여러 사업이 진행중이다. 이러한 사업을 통해 기존의 도시문제를 스마트시티만의 해결방법으로 해결하여 효율적인 도시관리를 가능하게 한다.

도시문제	기존도시 대응	스마트시티 대응	
에너지 부족	신규 인프라 건설	에너지효율화로 사용량 절감 네트워크 운영 최적화로 손실 절감	스마트 에너지
교통 혼잡	도로 확장, 건설	혼잡 정보 실시간 제공으로 우회 유도 실시간 교통량에 따른 교통신호 제어	스마트 교통
주차 문제	신규 주차장 건설	빈 주차공간 정보제공으로 주차 유도 카 셰어링 서비스로 도심진입 최소화	스마트 교통
방법, 안전	경찰 인력 확장	CCTV 정보공유로 경찰인력 즉각 투입 스마트범죄 앱 활용으로 도움 요청	스마트 복지
상하수도	누수지점 확인 불가	누수지점 센서 감지로 즉각 조치 가능 장기적 노후도 추정으로 누수 예측	스마트 환경

[그림2] Wi-SUN 최신동향

1.2. 기존 문제점

[그림3]을 보면 현재 국내 자동차 등록 수는 2000만대 이상으로 10년 전 대비 35% 가량 증가했다. 국민 약 2명당 한 대씩 보유하고 있다고 볼 수 있다. 하지만 주택가에 적절한 주차공간이 공급되지 않아 주차난이 점점 심해지고 있다. 그로 인해 주택가의 불법주차 문제가 지속적으로 발생한다. 또한 주택가 뿐만 아니라 관광지나 식당가처럼 인프라에 비해 주차공간 확보가 부족한 지역에서도 불법주차로 인한 민원과 사고가 끊이질 않고있다.

[그림3] 국내 자동차 등록대수

도시 교통 문제의 주요 원인을 제공하는 주차 문제는, 주차장 이용과 탐색에 과도하게 소모되는 시간과 주차고통(Parking Pain) 등 다양한 사회적 비용이 야기된다. INRIX Research의 조사에 따르면 주차로 인한 경제적 비용의 4분의 3 이상은 주차공간 탐색으로 인한 배회, 대기로 인한 비효율, 환경 문제에서 발생한다고 한다. 주차문제는 수요와 공급의 차이에서 발생한다. 출퇴근 시간, 휴일의 여행지 등 특정한 시간대나 시기에 수요가 몰리면서 주차장을 효율적으로 사용하기에 어려운 부분이 많다.

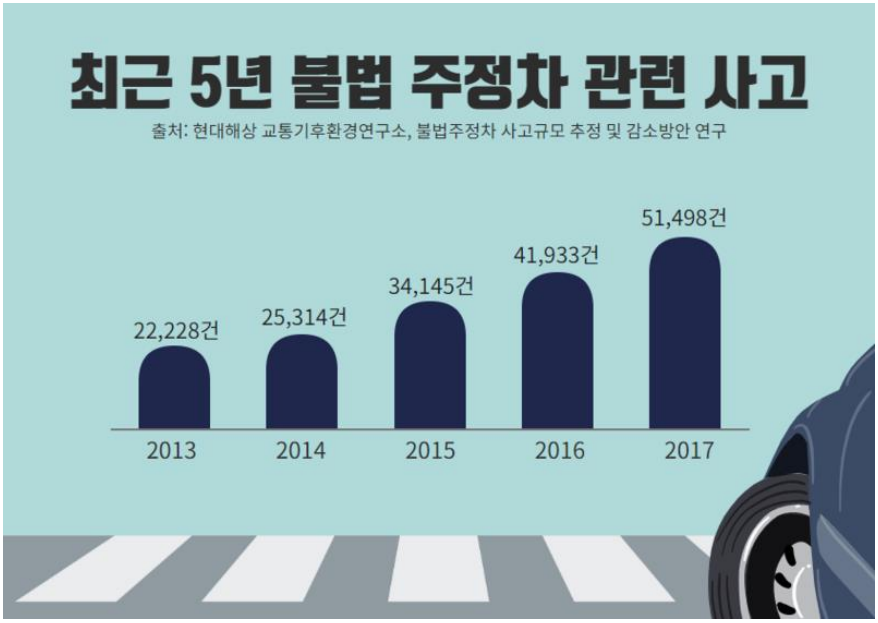
서울시의 불법 주정차로 인한 사회경제적 비용은 연간 4조 8970억원이다. 부천시는 개인 소유지나 공공시설의 주차공간을 여러사람이 사용할 수 있도록 개방하는 '공유주차' 서비스를 통해 주차공간을 확보하여 불법주차가 41%감소하는 효과를 거뒀다. 이처럼 우리가 제안하는 앱을 통해 도시의 사회경제적 비용을 줄이고자 한다.

1.3. 연구 목표

본 과제에서는 WI-SUN Mesh 모듈을 활용하여 도시의 주차장의 빈 자리 여부를 받아온 뒤 사용자에게 가장 적합한 주차장을 추천해주는 서비스를 제공한다.

모바일 어플리케이션을 통해 실시간으로 정보를 제공함으로써 불필요한 배회시간을 줄이고, 무선 네트워크를 활용한 주차장 연계 시스템을 구축하여 제한된 도시의 주차공간을 효율적으로 사용할 수 있으며 최소 인력으로 주차장 관리를 가능하도록 한다. 이를 통해 도시 내 곳곳에서 주차 공간 부족으로 발생하는 사회경제적 비용 문제와 불법 주정차 문제를 해결할 수 있다. [그림4]에서 알 수 있듯이 지속적으로 증가중인 불법 주정차 관련 사고를 예방하여 개인과 지자체의 손해를 줄일 수 있다.

실시간으로 수집한 데이터를 분석하여 이용률을 예상하고, 데이터를 시각화 하여 사용자가 한눈에 알아볼 수 있고 쉽게 이해할 수 있도록 한다. 이러한 시스템의 운영으로 쌓인 데이터를 통해 장기적인 주차 계획이나 향후 주차장 시스템 개선에 도움을 줄 수 있다.



[그림4] 최근 5년간 불법 주정차 관련 사고

2. 연구 배경

2.1. Wi-SUN Mesh 네트워크 선정이유

스마트 시티를 구성하는 네트워크는 지연시간이 낮고 안정성이 높아야 한다. 이를 위해 센서와 같은 소규모 장치의 저속통신을 위한 IoT기반 저전력 장거리 통신 기술인

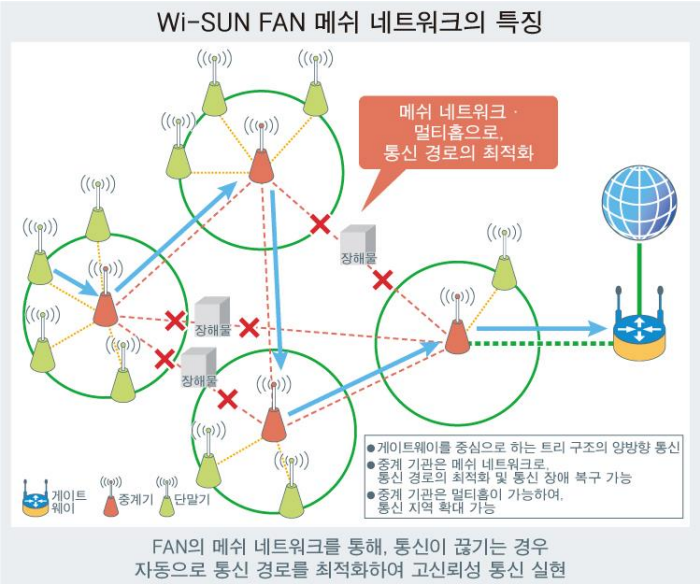
LPWAN(저전력 광역 통신망)을 사용한다. 이 중에서 스마트 시티에 적합한 프로토콜은 LoRaWAN, NB-IoT, Wi-SUN FAN이 있다.

이번 과제에 사용할 프로토콜은 Wi-SUN FAN이다. Wi-SUN은 Wireless Smart Utility Network의 약자로 비면허 대역을 사용하여 비용이 들지 않고, Mesh 네트워크를 통한 높은 확장성을 가지고 있으며, 단일 장애로 인한 시스템 다운을 방지하는 기능을 가진다. 또한 [그림5]의 비교표를 보면 LoRa, NB-IoT보다 전송속도는 높고 지연시간은 낮은 것을 알 수 있다. Wi-Fi 보다는 느리지만 통신거리가 길고 회절이 가능해 장애물에도 강하며 저소비 전력이라서 초대형 실외 Mesh 네트워크를 위한 통신 인프라를 제공하는 데 적당하다.

IoT Network	Data Rate	Latency
Wi-SUN FAN	Up to 300 Kbps	0.02 – 1 second
LoRaWAN	300 bps to 62.5 Kbps, depending on spreading factor	1-16 seconds
NB-IoT	Up to 140 Kbps Uplink, Up to 80 Kbps Downlink	2-10 seconds

[그림5] Wi-SUN FAN, LoRaWAN, NB-IoT 비교

Wi-SUN FAN을 활용한 무선 네트워크를 설치함으로써 야외 주차장과 같은 유선(배선) 설치가 어려운 공간에도 시스템 구축이 가능하며 설치 비용을 최소화할 수 있다.



[그림6] Wi-SUN FAN Mesh network 구성도

WI-SUN FAN은 향후 IoT 디바이스의 보급에 따라 대규모의 고신뢰성 네트워크 환경에서 이용될 것으로 기대하고 있다. 예를 들어 차세대 스마트미터, 스마트 가로등, 전기자동차 (EV)와 연동된 스마트 시티 및 인프라에서의 활용이 가능하다. 따라서 이번 과제에서 Wi-SUN FAN을 활용하여 스마트시티 주차장을 구현하고자 한다.

2.2. 주차장 문제 선정 이유

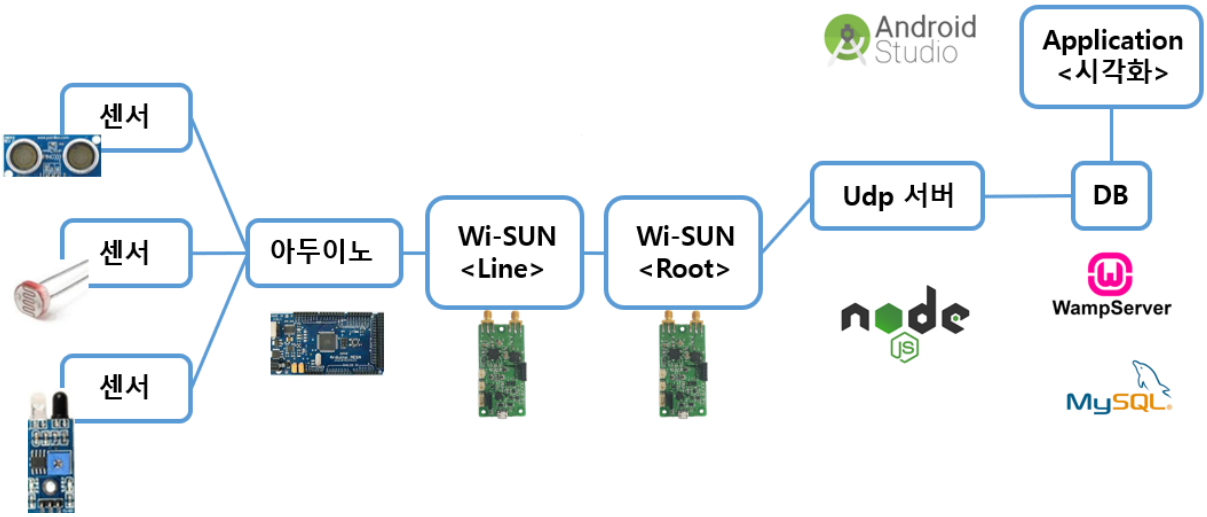
[그림7] '불법 주차를 한 이유' 설문조사 결과

기존 문제점에서 알 수 있듯이 늘어나는 자동차 수에 비해 부족한 주차장 문제를 해결할 수 있는 근본적인 방법이 주차장을 많이 확보하는 것이지만 현실적으로 불가능하므로 제한된 면적에서 효율적으로 주차 공간을 확보하는 것이 중요하다. 네트워크 구축을 통해 도시를 스마트하게 관리한다면 이 문제를 효과적으로 해결할 수 있다고 생각하였다.

또한 앞서 본 [그림2]의 스마트 시티의 기존 문제점 대응들 중에서 가장 직관적으로 확인 가능하고 여러 센서와 아두이노 보드, Wi-SUN 모듈 간의 연동을 통해서 효과적인 관리 시스템을 구축할 수 있는 주제라고 생각하여 주차장 문제해결을 위한 시스템을 구현하고자 한다.

3. 연구 내용

3.1. 네트워크 구성

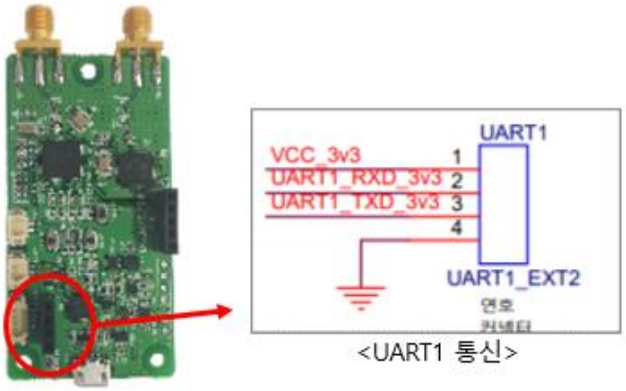


[그림8] Wi-SUN 모듈과 UART1 포트 구성도

총 12칸의 주차공간을 가정하여 구현하였고, 각 주차공간마다 초음파 센서, 조도 센서, 적외선 센서를 설치했다. 주차 유무의 정확성을 높이기 위해 초음파 센서와 적외선 센서가 동시에 주차라고 판단하면 해당 자리는 주차된 자리라고 판단하도록 구현했고, 조도 센서는 해당 자리가 그늘진 자리인지 원하는 사람을 위해 그늘진 자리 유무 판단을 위해 추가했다.

12칸을 4칸씩 3줄로 나눠서 각 줄마다 아두이노와 모듈을 연결했다. 아두이노 하나 당 총 12개의 센서가 연결되는 구조이다. 12개의 디지털 핀과 4개의 아날로그 핀을 사용했고 아두이노와 Wi-SUN 모듈을 UART1 통신으로 연결되어 있다. 한 주차장 내에서 각 줄의 정보를 받아오는 Wi-SUN 모듈이 각 주차장의 정보를 한 번에 모으는 Wi-SUN모듈로 정보를 전달하고 그 주차장 Wi-SUN모듈은 모은 정보를 UDP패킷 형태로 서버에 전달한다. Nodejs로 구현한 UDP서버에서 센서의 정보를 받아서 정보를 파싱하여 DB에 주차장 고유번호와 줄 번호, 그리고 자리에 대한 정보를 0,1,2의 값으로 입력한다. 어플리케이션에서는 각 정보를 읽어서 0이면 햇빛+자리 있음, 1이면 그늘+자리 있음, 2이면 자리 없음을 표시해준다.

3.1.1. 드라이버 구현



[그림9] Wi-SUN 모듈과 UART1 포트 구성도

드라이버는 Wi-SUN 모듈에서 UART1핀을 사용하여 통신하도록 구현하였고, 역할을 보면 아두이노에서 5초마다 주차장 정보를 HDLC frame에 담아 송신하면 모듈에서는 해당 frame에 들어있는 정보를 inputbuffer에 저장한다. 이 정보를 UDP 패킷에 담아서 모듈이 속한 네트워크의 루트를 향해 정보를 전달한다. 이 때 만약 해당 모듈이 루트 모듈이면 UDP패킷을 서버로 전달한다. 서버가 읽고 있는 포트를 목적지 포트로 설정하여 전달하면 해당 포트를 목적지로 하는 패킷을 전부 서버가 수신하게 된다.

```
outputHdlcOpen();
outputHdlcWrite(SERFRAME_MOTE2PC_DATA);
outputHdlcWrite((uint8_t)'1');//주차장 고유번호
outputHdlcWrite((uint8_t)'1');//주차장 줄 고유번호
outputHdlcWrite((uint8_t)p1);//주차자리 상태
outputHdlcWrite((uint8_t)p2);
outputHdlcWrite((uint8_t)p3);
outputHdlcWrite((uint8_t)p4);
outputHdlcClose();
```

[그림10] 아두이노에서 전달하는 데이터 형식

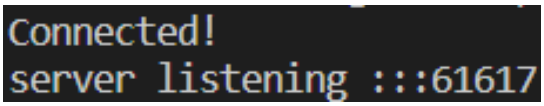
아두이노에서 오는 주차장 정보는 [그림10]와 같다. 주차장 고유번호와 해당 모듈의 줄 번호가 오고 이후로 각 자리의 현재 주차현황을 각각 p1, p2, p3, p4에 담아 전달한다.

3.2. DB 및 서버 구축

- UDP서버

```
var sql = "INSERT INTO park_info_8 (park_num, park_line, park_1st, park
          "ON DUPLICATE KEY UPDATE park_1st = " + "'" + park_1st + "',
```

[그림11] UDP 서버 SQL문 일부



[그림12] UDP 서버 연결 확인

[그림13] UDP 서버 연결 확인

UDP서버는 Nodejs로 구현하였고, Root모듈에서 보낸 String 데이터가 미리 맞춰 놓은 포트인 61617번 UDP 서버로 들어온다. 이 값을 받은 후 int값으로 변환하여 SQL문을 통해 SQL문을 이용하여 DB에 각 컬럼에 맞게 입력한다. DB에 데이터가 알맞게 들어가면 [그림 12]와 같은 문구를 통해 확인할 수 있다.

- DB

Wampserver를 통해 MySQL을 실행한다. MySQL은 UDP서버로부터 정보를 받아서 [그림 13]와 같은 형태로 저장한다. 여기서 주차장 번호, 주차장 라인을 Key값으로 하여 새로 들어온 데이터와 두 Key값이 같은 데이터가 이미 있으면 덮어쓰기 하여 주차장의 각 라인마다 데이터가 하나씩만 유지되도록 하였다. 각 자리의 값이 0이면 햇빛이 들어오며 주차가 가능한 자리, 1이면 햇빛이 들어오지 않으며 주차가 가능한 자리, 2이면 주차가 불가능한 자리를 의미한다.

주차장 번호	주차장 라인	1번 자리	2번 자리	3번 자리	4번 자리
1	1	0	0	1	2
1	2	2	2	2	2
1	3	1	2	2	2

[그림14] Table 형태 예시

3.3. 어플리케이션 구현

3.3.1. 사용 도구

- [안드로이드 스튜디오]

  Android 스튜디오는 Android 앱 개발을 위한 공식 통합 개발 환경(IDE)이며 IntelliJ IDEA를 기반으로 한다. IntelliJ의 강력한 코드 편집기와 개발자 도구 외에도, Android 스튜디오는 Android 앱을 빌드할 때 생산성을 높여주는 기능을 제공한다. 어플리케이션의 이름은 “여기 댈래?”로 정하였고, [그림15]와 같이 주차장 모양의 아이콘을 선택했다.

[그림15] 어플리케이션 아이콘

- [사용 라이브러리]

  OKhttp : 효율적인 http 클라이언트로, 쉽게 request/response를 할 수 있도록 도와주는 오픈소스 라이브러리이다. 외부의 데이터를 받아 오기 위해 사용했다.

```
/* Open api 호출 (주차장) */
private void callOpenApi() {
    try {
        // 오픈 api 호출
        OkHttpClient okHttpClient = new OkHttpClient.Builder()
                .connectTimeout( timeout: 60, TimeUnit.SECONDS)
                .writeTimeout( timeout: 60, TimeUnit.SECONDS)
                .readTimeout( timeout: 60, TimeUnit.SECONDS)
                .build();
```

  Karumi Dexter : 안드로이드 앱 권한 요청을 쉽게 할 수 있도록 도와주는 라이브러리이다.

  smart-fun.XmlToJson : XML을 JSON으로, JSON을 XML로 쉽게 변환하는 라이브러리이다.

- [API (Application Programming Interface)]

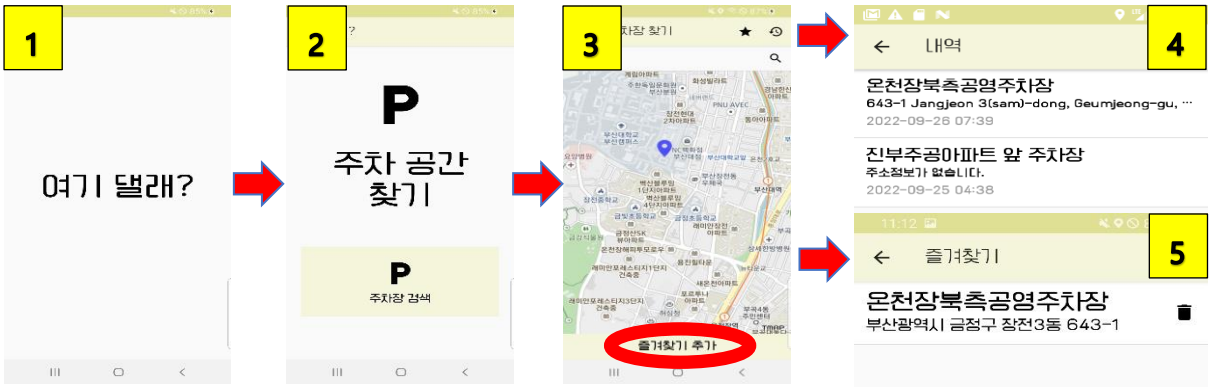
응용 프로그램 프로그래밍 인터페이스. 프로그래밍에서, 프로그램을 작성하기 위한 일련의 부(Sub) 프로그램, 프로토콜 등을 정의하여 상호 작용을 하기 위한 인터페이스 사양을 말한다. 데이터 공유가 필요한 경우 API에 정보를 요청하며, API는 Key 값을 개별적으로 발급해 특정 서비스를 이용할 수 있게 한다. Key 값은 데이터에 대한 액세스 권한을 주어 악의적 사용 혹은 남용을 방지한다.

공공데이터포털은 공공기관이 생성 또는 취득하여 관리하고 있는 공공데이터를 한 곳에서 제공하는 통합 창구이다. 포털에서는 국민이 쉽고 편리하게 공공데이터를 이용할 수 있도록 파일데이터, 오픈API, 시각화 등 다양한 방식으로 제공하고 있으며 누구든지 쉽고 편리한 검색을 통해 원하는 데이터를 빠르고 정확하게 찾을 수 있다.

TMAP Open API는 지도를 기반으로 하는 다양한 서비스를 지원한다. 지도 서비스뿐만 아니라 경로안내, 주소검색, 명칭(POI) 검색, 공간검색, 실시간 교통정보 검색 등 다양한 기능을 Restful API 기반으로 제공하고 있다.

WI-SUN을 이용해 빈 자리가 있는 주차장을 찾아가는 것이 목표이므로, 주차장 데이터와 TMAP API를 사용했고 API Key 값을 얻어와 Android Studio의 strings.xml에 추가했다.

3.3.2. UI 설명



- 1. Intro Activity : 애플리케이션 실행 시 앱 이름이 뜬다.

- 2. Main Activity : '주차 공간 찾기'라는 간단한 문구가 보이고, 그 아래에 두 가지 activity로 넘어갈 수 있는 Button이 있다. 좌측은 '주차장 찾기', 우측은 '주차 현황'이다.

- 3. National Parking Lot Activity : 좌측 '주차장 검색'을 누르면 사용자의 GPS 마크가 현재 위치한 지도에 표시되고, 지도는 손가락 터치로 확대, 축소, 이동이 가능하다. 상단바에 주차장 찾기 문구가 있고, 그 왼쪽에는 '뒤로 가기' 버튼이, 오른쪽에는 '즐겨찾기'와 '검색기록' 버튼이 있다. 그 아래에 돋보기 모양을 누르면 '주차장검색' activity로 넘어갈 수 있다. 그리고 지도 하단에 '즐겨찾기 추가'가 있어, 해당 주차장을 즐겨찾기 목록에 추가할 수 있다. 검색 기록과 즐겨찾기의 경우, 아래 코드처럼 SQLite를 이용해 구현했다.

- 4. History Activity : 주차장을 검색하고 상세 정보를 확인했을 때, 해당 주차장은 검색 내역에 추가된다. 추가된 주차장은 이후에 따로 검색을 하지 않아도 바로 주차장 상세 정보를 확인하거나 길찾기 기능을 이용할 수 있다. 검색 내역 역시 목록에서 삭제가 가능하다.

-

- 5. Bookmark Activity : Main Activity 상단바의 별 모양 아이콘을 누르면, 현재까지 즐겨찾기 추가한 주차장 목록을 볼 수 있다. 목록에 있는 주차장을 클릭하면 주차장 상세 정보를 볼 수 있고 길찾기 기능을 이용할 수 있으며, 목록에서 삭제도 가능하다.

- 6. National Parking Lot Search Activity : 공공데이터포털에서 제공하는 전국의 주차장 데이터를 검색할 수 있다. 주차장은 운영 방식, 유형, 요금에 따라 나누어 검색 가능하다.

- 7. Parking Lot Info Activity : 주차장 검색 화면에서 원하는 주차장을 클릭 하면, 해당 주차장의 상세 정보를 확인할 수 있는 activity로 넘어간다. 먼저 주차장의 이름과, 현재 사용자의 GPS 위치에서 해당 주차장까지의 거리 정보가 상단에 뜬다. 그 아래에 공공데이터 포털에서 제공하는 상세 정보가 나열되는데, (주차장 관리번호, 주차장 구분, 유형, 주소, 주차 구획 수, 운영 요일, 시간, 전화번호 등)의 정보가 표시된다.

- 8. 길 찾기 기능 : 7번 화면에서 길찾기 버튼을 누르면 현재 위치에서 주차장까지의 최단 경로가 나온다. 이때 Tmap API를 사용하여 최단 경로를 계산한다.

- 9. Database Activity : Main Activity의 '주차 현황'을 누르면 위 activity로 넘어온다. 주차장에 설치된 센서와 아두이노의 정보를 Wi-SUN Mesh 네트워크로 받아오고, 서버를 통해 데이터베이스에 저장한다. 이 저장된 값을 해당 activity에서 확인할 수 있는데, 주차장의 온도, 조도, 주차 현황을 볼 수 있다. 주차 칸에 차가 있다고 인식되었으면 빨간색, 차가 없다고 인식되었으면 파란색을 띄게 했다.

```
int park_1st_int = Integer.parseInt(mList.get(position).getPark_1st());
int park_2nd_int = Integer.parseInt(mList.get(position).getPark_2nd());
int park_3rd_int = Integer.parseInt(mList.get(position).getPark_3rd());
int park_4th_int = Integer.parseInt(mList.get(position).getPark_4th());

if(park_1st_int == 2){
    viewholder.park_1st.setText(" 1 ");
    // 입차
    viewholder.park_1st_f.setBackgroundResource(R.color.red_icon_color);
    viewholder.park_1st_f.setVisibility(View.VISIBLE);
}
else if (park_1st_int == 1) {
    viewholder.park_1st.setText(" 1 ");
    // 입차
    viewholder.park_1st_f.setBackgroundResource(R.color.grey);
    viewholder.park_1st_f.setVisibility(View.VISIBLE);
}
else{
    viewholder.park_1st.setText(" 1 ");
    // 출차
    viewholder.park_1st_f.setBackgroundResource(R.color.blue_icon_color);
    viewholder.park_1st_f.setVisibility(View.VISIBLE);
}
```

데이터베이스에서 앱으로 php 파일을 불러올 때, PersonalData.java, DBHelper.java, UsersAdapter.java 등의 파일을 추가했다. 데이터베이스에서 불러온 값을 읽어, 현재 주차장 상태를 표시했다.

```
<androidx.recyclerview.widget.RecyclerView
    android:id="@+id/listView_main_list"
    android:layout_width="match_parent"
    android:layout_height="0dp"
    android:layout_margin="3dp"
    android:layout_weight="6"
    android:fontFamily="@font/cafe24"
    android:textColor="@color/black"/>
```

화면에 표시할 때는 item_list.xml 에 넣어둔 데이터베이스 값을, activity_db.xml 의 RecyclerView 를 통해 보여주는 방식으로 Layout 을 구현했다.

4. 결론 및 향후 연구 방향

센서로 받아온 정보를 여러 통신방식을 통해 어플리케이션에서 실시간으로 확인 가능하도록 구현하여 기존에 생각했던 구성대로 연결에 성공하였다. uart통신, udp통신 등 여러 통신 방법을 사용해보면서 이론으로만 배웠던 통신에 대해 조금 더 이해할 수 있었고, 학부과정에서 여러 실험을 통해 배운 아두이노, 안드로이드 스튜디오 등을 연동하여 결과물을 완성하면서 이전에 했던 과목들에 대한 지식을 다시 한번 상기시켜주는 경험이었다.

비록 본 과제에서는 공간 협조나 센서 부족의 문제로 실제 주차장에서 테스트를 해보지 못했지만 현재 구현한 부분에서 모듈과 센서, 공간의 문제만 해결된다면 시스템이 쉽게 확장될 수 있다고 생각한다. 추가적으로 현재는 루트 모듈에서 모든 정보를 직접 받지만 시스템이 확장된다면 모듈을 트리형태로 연결하여 좀 더 효율적으로 정보를 관리할 수 있다고 생각한다. 또한 추후 이 시스템의 사용자가 많아지고 오랜 기간동안 데이터를 모은 다음 수집한 데이터를 잘 활용한다면 해당 도시의 주차공간 확보 계획에 도움을 줄 수 있을 것으로 기대한다.