

삼성 오픈소스 컨퍼런스

내손안의 수질연구소

수질 공공데이터를 활용한
안드로이드 기반 애플리케이션

영남중학교 | 김민겸, 김지환, 조현성
2019.10.17



주제 및 문제점

문제 상황 및 제작 동기 제시



SOSCON 2019

SAMSUNG OPEN SOURCE CONFERENCE 2019

대구 수돗물 발암 물질 검출 사태

대구 매곡/문산 정수장
과불화화합물 검출 사태
2018. 6. 22

VS

대구시 상수도 본부, 환경부
과불화화합물 문제없음 발표
2018. 6. 27

낙동강 페놀 검출 사태
(1991.03.14)



낙동강 페놀 검출 사태
(2008.03.01)



문제를 인식하게 된 계기



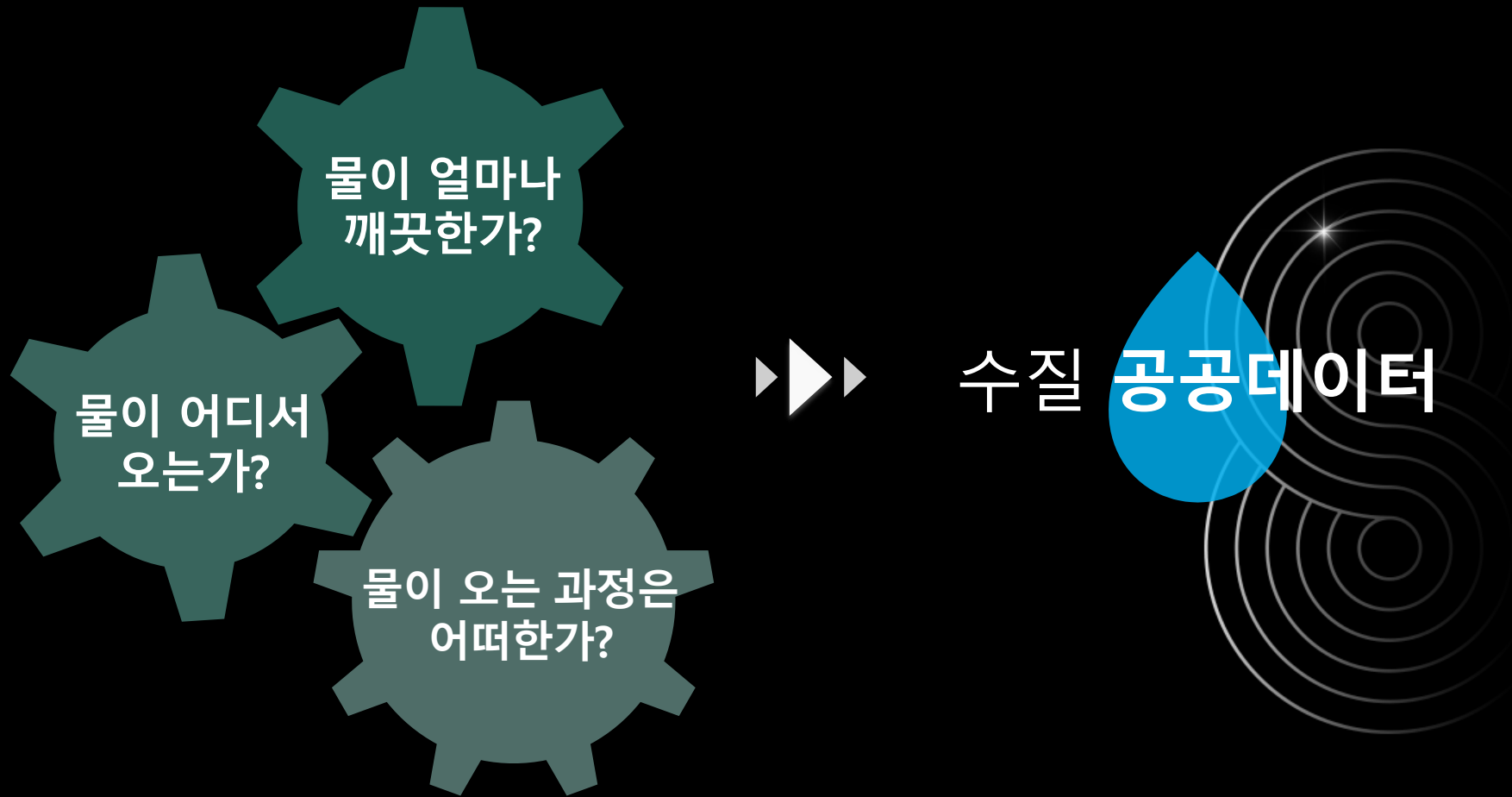
수질 오염
발생 횟수의 증가



수질 문제 관련
사람들의 관심 증가



수질 문제에
대한 불안감

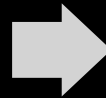




시간의 제약



해석의 제약



공공데이터 자체로는 시민들이
수질 정보를 확인할 수 없음

문제 해결 방법



- 실시간으로 사용자의 위치를 확인, 근처 정수장의 지수를 보여줌
- 수질 단계에 따라 화면 색을 다르게 하여 눈에 잘 띄도록 구성
- 수질이 위험 단계일 시 알림으로 알려줌



- 여러 수질 데이터를 종합해 하나의 간결한 수치와 단계로 표현함
- 생활에 필요한 실용적인 데이터 또한 제공함으로써 사용자들이 데이터를 해석할 필요가 없도록 개발

제작 과정

앱 설계 및 구현 과정 소개



공공데이터 탐색 및 활용 신청

```
<fcityMngNm>사천정수장</fcityMngNm>
<fcityMngNo>4824012333</fcityMngNo>
<liIndDivName>생활정수</liIndDivName>
<no>2</no>
실시간
수질정보
<ocrrncDt>2019100315</ocrrncDt>
<phUnit>PK</phUnit>
<phVal>6.972</phVal>
<tbUnit>NTU</tbUnit>
<tbVal>.0491</tbVal>
</item>
```



pH



탁도



잔류염소

```
<item>
<item1>무미</item1>
<item2>무취</item2>
<item3>0.03</item3>
광역정수장 수질
(일일)
<item5>0.8</item5>
<mesurede2>111</mesurede2>
</item>
<item>
<item1>무미</item1>
<item2>무취</item2>
```



맛



냄새



색도

```
<item>
<item1>불검출</item1>
<item2>불검출</item2>
<item3>불검출</item3>
<item4>불검출</item4>
<item5>1.7</item5>
<item6>1.0</item6>
<item7>280</item7>
<mesurede>20181004</mesurede>
</item>
```



세균/대장균



KMnO4소비량



증발잔류물

N₂

암모니아성/질산성질소



알고리즘 및 코드

완성된 앱의 알고리즘 및 소스코드

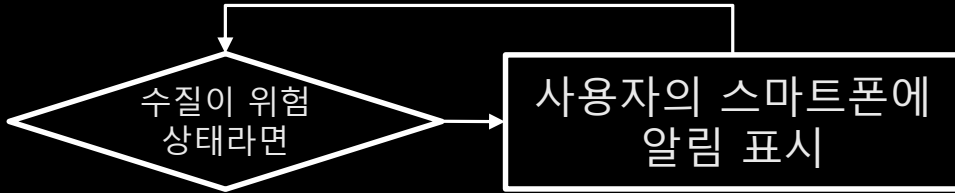


SOSCON 2019

SAMSUNG OPEN SOURCE CONFERENCE 2019

알고리즘 순서도

Service Background



App Activity



소스코드 – MainActivity

권한 요청

```
private boolean hasPermissions() {  
    int res = PackageManager.PERMISSION_DENIED;  
  
    String[] permissions = new String[]{Manifest.permission.ACCESS_FINE_LOCATION, Manifest.permission.ACCESS_COARSE_LOCATION, Manifest.permission.WRITE_EXTERNAL_STORAGE, Manifest.permission.READ_EXTERNAL_STORAGE};  
  
    for (String perms : permissions) {  
        res = checkCallingOrSelfPermission(perms);  
  
        if (!(res == PackageManager.PERMISSION_GRANTED))  
            return false;  
    }  
    return true;  
}  
  
private void requestPerms() {  
    String[] permissions = new String[]{Manifest.permission.ACCESS_FINE_LOCATION, Manifest.permission.ACCESS_COARSE_LOCATION, Manifest.permission.WRITE_EXTERNAL_STORAGE, Manifest.permission.READ_EXTERNAL_STORAGE};  
  
    if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.M) {  
        requestPermissions(permissions, requestCode);  
    }  
}
```

안드로이드 마시멜로우 이상부터는 위험권한 획득을 위한 권한 요청이 필요하다.

- 권한이 승인된 경우 false를, 그렇지 않은 경우 true를 반환
- 안드로이드 마시멜로우 이상의 경우 권한 요청 메시지를 보냄

소스코드 - MainActivity

GPS 좌표 및 정수장과의 거리 계산

```
double x = Double.parseDouble(getString(getStringIdentifier(getApplicationContext(), name: "fcityX" + Integer.toString(i))));
double y = Double.parseDouble(getString(getStringIdentifier(getApplicationContext(), name: "fcityY" + Integer.toString(i))));
String name = getString(getStringIdentifier(getApplicationContext(), name: "fcityNm" + Integer.toString(i)));

//double distance = Math.sqrt(Math.pow(lat-x, 2)+Math.pow(lng-y, 2));
double distance = 40075 / 360 * Math.acos(Math.cos(90 - lat) * Math.cos(90 - x) + Math.sin(90 - lat) * Math.sin(90 - x) * Math.cos(Math.abs(lng - y)));

for (int j = 0; j < 5; j++) {
    if (distance < distancelist.get(j)) {
        locate = j;
    } else {
        break;
    }
}
```

정수장의 x, y 좌표를 받아온 뒤 현 GPS 좌표와 차이를 계산한다.

- 구면 코사인 법칙을 통해 각각의 정수장과의 거리를 산정
- 가장 가까운 근처 5개 정수장을 확인

소스코드 - MainActivity

시간 확인 및 스레드 시작

```
LocationListener locationListener = new LocationListener() {  
    public void onLocationChanged(Location location) {  
        long now = System.currentTimeMillis();  
        Date date = new Date(now);  
        Calendar cal = Calendar.getInstance();  
        gettime = sdf2.format(date);  
        if(dest!=1 && hour!=Integer.parseInt(gettime))  
        {  
            dest = 1;  
            hour = Integer.parseInt(gettime);  
  
            score = new ArrayList<Integer>(Arrays.asList( 1, -1, -1, -1, -1));  
            getdate = sdf.format(date);  
            getweek = sdf3.format(date);
```

```
        if (gettime.equals("24")) {  
            cal.add(cal.DATE, amount -1);  
            getdate = sdf.format(cal.getTime());  
        }  
        try {  
            CaIThread thread = new CaIThread();  
            thread.start();  
        } catch (Exception ignored) {  
        }  
    }  
}
```

- GPS 좌표 변동 시 현재 시간/날짜를 가져옴
- 네트워크 작업은 별도의 스레드가 필요하기 때문에 스레드를 만들어 실행

소스코드 – MainActivity

데이터 파싱 및 수치 계산

```

if (Mcl.find()) {
    ciValIS = Mcl.group(0).substring(7, Mcl.group(0).length() - 8);
    if (ciValIS.charAt(0) == '.') {
        ciValIS = "0" + ciValIS;
    }
    ciValList.add(k, ciValIS);
    ciValList.remove(index: k + 1);
    ciVal = Float.parseFloat(ciValIS);
}
if (Mph.find()) {
    phVal = Float.parseFloat(Mph.group(0).substring(7, Mph.group(0).length() - 8));
    phValList.add(k, Mph.group(0).substring(7, Mph.group(0).length() - 8));
    phValList.remove(index: k + 1);
}
if (Mtb.find()) {
    tbValIS = Mtb.group(0).substring(7, Mtb.group(0).length() - 8);
    if (tbValIS.charAt(0) == '.') {
        tbValIS = "0" + tbValIS;
    }
    tbValList.add(k, tbValIS);
    tbValList.remove(index: k + 1);
    tbVal = Float.parseFloat(tbValIS);
}

```

```

if (!nbVal.equals("불검출") && !nbVal.equals("")) {
    if (Float.parseFloat(nbVal) > 15) {
        ++ofn;
        sum += Float.parseFloat(nbVal) / 15 - 1;
    }
}
if (!nmVal.equals("불검출") && !nmVal.equals("")) {
    if (Float.parseFloat(nmVal) > 0.075) {
        ++ofn;
        sum += Float.parseFloat(nmVal) / 0.075 - 1;
    }
}
if (nnVal > 1.5) {
    ++ofn;
    sum += nnVal / 1.5 - 1;
}
if (brVal > 75) {
    ++ofn;
    sum += brVal / 75 - 1;
}
float f1 = (float) ofn / (float) num * 100;
float f3 = (sum / (float) num) / ((float) 0.01 + (sum / (float) num) + (float) 0.01);
score.add(k, (int) Math.round(Math.sqrt((Math.pow(f1, 2) * 2 + Math.pow(f3, 2)) / 3)));
score.remove(index: k + 1);

```

- 가져온 데이터를 적절히 파싱하여 변수에 저장
- 수식에 대입하여 수질 수치 및 단계를 도출



소스코드 – MainActivity

계산된 수질 점수 적용

```
if (totalscore < 20) {  
    title.setText("매우 좋음");  
    desc.setText("안심하고 물을 드셔도 되겠네요!");  
    toolbar.setBackgroundColor(getResources().getColor(R.color.colorPrimary));  
    window.setStatusBarColor(ContextCompat.getColor(context, MainActivity.this, R.color.colorPrimaryDark));  
    wave.setImageResource(R.drawable.wave_verygood);  
}  
else if (totalscore < 40) {  
    title.setText("좋음");  
    desc.setText("수질이 안전한 단계입니다.");  
    toolbar.setBackgroundColor(getResources().getColor(R.color.colorGood));  
    window.setStatusBarColor(ContextCompat.getColor(context, MainActivity.this, R.color.colorGoodDark));  
    wave.setImageResource(R.drawable.wave_good);  
}  
else if (totalscore < 60) {  
    title.setText("보통");  
    desc.setText("평범한 날이네요.");  
    toolbar.setBackgroundColor(getResources().getColor(R.color.colorNormal));  
    window.setStatusBarColor(ContextCompat.getColor(context, MainActivity.this, R.color.colorNormalDark));  
    wave.setImageResource(R.drawable.wave_normal);  
}
```

- 계산된 수질 단계 / 점수를 표시
- 수질 단계에 따라 화면 색을 조정함



예측 및 그래프 표시

```
int mis;  
int best = 700;  
int besti=0;  
int bestj=0;  
for(int i=0;i<=100;i++){  
    for(int j=-20;j<=20;j++){  
        mis=0;  
        for(int m=0;m<6;m++){  
            mis+=Math.abs(prevScore.get(m)-(j+m+1));  
        }  
        if(mis<best){  
            best=mis;  
            besti = i;  
            bestj = j;  
        }  
    }  
}  
int pre = bestj*6+besti;  
if (pre>100)pre=100;  
else if (pre<0)pre=0;  
lineEntries.add(new Entry( x ?f, pre));
```

- 최근 6시간 동안의 수질 점수, 간단한 선형회귀를 통한 예측 데이터 그래프로 표시



알림 보내기

```
if (totalScore >= 60 && notice.equals("T")) {
    Intent intent = new Intent(packageContext, MyService.this, MainActivity.class);
    PendingIntent pendingIntent = PendingIntent.getActivity(context, MyService.this, requestCode, 0, intent, PendingIntent.FLAG_UPDATE_CURRENT);

    //PendingIntent pendingIntent = PendingIntent.getActivity(this, 0, new Intent(this, MainActivity).0);
    NotificationCompat.Builder builder = new NotificationCompat.Builder(context, MyService.this, channelId: "default");

    builder.setSmallIcon(R.mipmap.ic_launcher);
    builder.setDefaults(Notification.DEFAULT_VIBRATE);
    builder.setSmallIcon(R.drawable.icon);
    builder.setContentTitle("근처 정수장의 수질이 위험 상태입니다!");
    builder.setContentText(String.valueOf(totalScore) + "%/100");
    builder.setPriority(NotificationCompat.PRIORITY_HIGH);
    builder.setContentIntent(pendingIntent);

    builder.setAutoCancel(true);

    // 알림 표시
    NotificationManager notificationManager = (NotificationManager) getSystemService(Context.NOTIFICATION_SERVICE);
    if (notificationManager != null) {
        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.O) {
            notificationManager.createNotificationChannel(new NotificationChannel("default", "기본 채널", NotificationManager.IMPORTANCE_DEFAULT));
        }
        notificationManager.notify(1, builder.build());
    }
}
```

- 앱이 꺼져 있더라도 서비스가 실행 중
- 수질이 위험 상태일 경우 스마트폰에 알림 보냄



화면 구성

완성된 앱의 실제 동작 화면



메인 화면



수질 단계

계산된 수질 수치

근처 정수장 5개의
거리와 상태

세부 정보 버튼



세부정보 화면



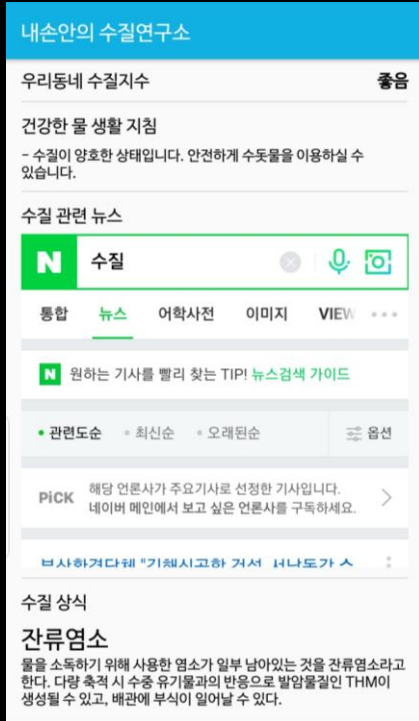
정수장 선택

세부정보

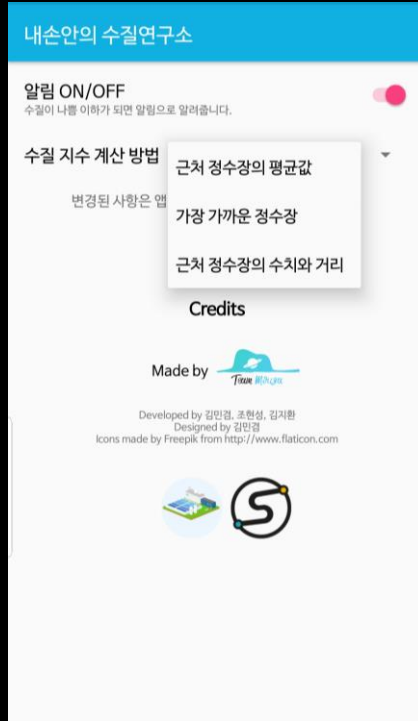
날짜별 비교 및
예측 그래프



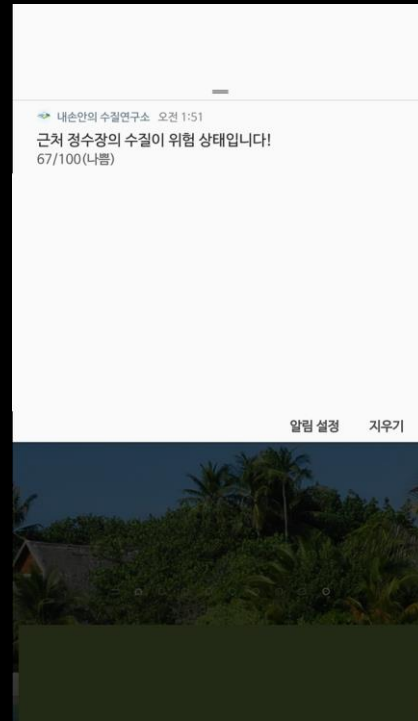
생활정보 화면



설정 화면



알림 화면



화면 구성

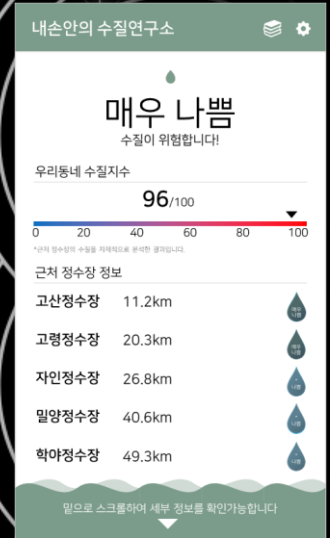
매우
*
중음

중음
*
중음

보통
*
보통

나쁨
+
나쁨

매우
+
나쁨





장점 및 개선 방안

프로젝트의 가치와 향후 지향 목표



SOSCON 2019

SAMSUNG OPEN SOURCE CONFERENCE 2019



적합성

가까운 정수장의 수질 정보를 제공



편의성

각 수질 단계에 대한 생활 지침까지 제공



전달성

이해하기 쉽게 색/그래프로 표시



한계점 및 개선 방안

한계점



수도배관에서 수질오염 발생 시 확인 불가



GPS 좌표로는 내가 먹는 물을 공급하는 정수장을 확인할 수 없음

개선 방안

집에 센서를 달아 직접 데이터를 수집한다. (추가 서비스)

정수장이 물을 제공하는 **구역정보**를 통해 사용자에게 어느 정수장이 물을 제공하는지 알려준다.

THANK YOU

SOSCON 2019

SAMSUNG OPEN SOURCE CONFERENCE 2019

