안드로이드 기반 버스알람 시스템

차주운*, 이진현, 공기석
한국산업기술대학교 컴퓨터공학과
e-mail : {goldman556*, ljhboss, kskong}@kpu.ac.kr

Android-based Bus Destination Alarm System

Ju-Un Cha, Jin-Hyun Lee, Ki-Sok Kong
Department of Computer Engineering, Korea Polytechnic University

요 약

사용자가 버스정류장과 알람반경을 선택하면 미리 설정한 옵션에 따라 소리 또는 진동으로 사용자에게 알려준다. 안드로이드 2.1 플랫폼에서 GPS센서와 SQLite3 DataBase, Wifi 등을 컨트롤 할 수 있는 API 들을 사용하여 개발하였다.

1. 서론

최근 세계 스마트폰 시장이 급격한 출하량 증가를 보이고 있다[1]. 이에 따라 스마트폰 운영체제들 또한 주목을 받고 있다. 이 중 아이폰과 안드로이드 운영체제가 대표적이며, 안드로이드가 아이폰의 점유율을 점차 따라잡고 있는 상황이다[2].

스마트폰에는 기존 휴대폰과 달리 GPS, Wifi, 속도계 등 다양한 센서들이 탑재되어 있다. 따라서 이런 센서들을 이용하면 일상생활에 유용하거나 흥미로운 애플리케이션들을 만들 수 있다.

버스알람을 만들게 된 계기는 단순하다. 예전에 버스를 타고 가다가 피곤한 나머지 잠들어서 내려야 할 정류장을 지나쳐 버린 적이 있었다. 그 후로 마음 편안하게 버스에서 잠잘 수가 없었는데, 누군가 도착했을 때 깨워주면 좋겠다고 생각했다. 본 어플리케이션은 사람들의 일상을 좀 더 편안하게 바꿔줄 수 있는 도구 중 하나가 될 것이다.

2. 관련연구

2.1 기존 유사 앱과의 비교

아래의 [표 1]은 본 논문의 앱과 기존의 유사 앱들 간의 기능에 대해서 비교하고 있다[3][4].

<표 1> 기존 유사 앱과 기능비교

	WakeUpnow	GPS Alarm	버스알람
소리와 진동설정	O	O	O
알람 반경 설정	O	O	O
배터리 부족 알람	X	X	O
서울버스노선내장	X	X	O
남은 거리 표시	X	X	O

본 애플리케이션은 최초 한번만 노선정보를 다운로드하면 이후에는 목적지 설정부터 알람까지 인터넷 접속이 필요 없기 때문에 비용에 부담이 없다. 간혹 배터리가 부족해서 알람 전에 핸드폰이 꺼지거나, GPS 신호를 수신할 수 없는 지역으로 들어가서 현재위치의 변화가 없는 경우가 있을 수 있다. 이에 대비하여 옵션으로 배터리가 10% 미만일 경우 알림 옵션과 1분 이상 현재위치가 바뀌지 않을 경우 알려 주는 옵션을 만들었다. 만약에 있을지 모르는 경우에도 최소한 목적지 전에는 깨우기 위해 고민했다.

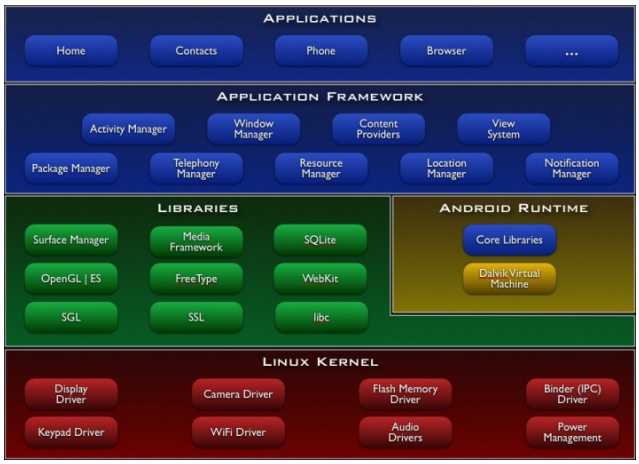
2.2 안드로이드 플랫폼

안드로이드는 운영체제와 미들웨어 그리고 핵심 애플리케이션을 포함하고 있는 모바일 디바이스를 위한 소프트웨어 스택Stack이다. 안드로이드 SDK는 Java 프로그래밍 언어를 사용하여 안드로이드 플랫폼 상의 애플리케이션을 개발하기 위한 도구들과 API를 제공한다[5][6].
[표 2]는 안드로이드에서 지원하는 여러 기능들을 소개하고 있다.

<표 2> 안드로이드 플랫폼의 특징

특 징	설 명
Application framework	컴포넌트를 재사용하거나 대체가능
Dalvik virtual machine	모바일 장치에 최적화된 가상머신
browser	open source WebKit engine 기반
2D,3D graphics library	OpenGL ES 1.0 기반의 3D graphics
SQLite 데이터베이스	모바일 장치를 위한 가벼운 저장공간
미디어 지원	일반적인 오디오, 비디오 포맷 지원
Camera, GPS, compass, and accelerometer	하드웨어에 의존적인 센서들
풍부한 개발환경	에뮬레이터, 디버깅 툴, Eclipse Plug-in

[표 2]에서 볼 수 있듯이 안드로이드 플랫폼은 미디어 지원, 감지센서, 그래픽 라이브러리 등을 지원한다. 이런 기능들은 [그림 1]에서와 같이 애플리케이션 프레임워크가 하위구조와 애플리케이션 사이에 존재함으로써 간단한 API 만으로 하위구조와 상관없이 쉽게 조작할 수 있다.



(그림 1) 안드로이드 플랫폼의 구조

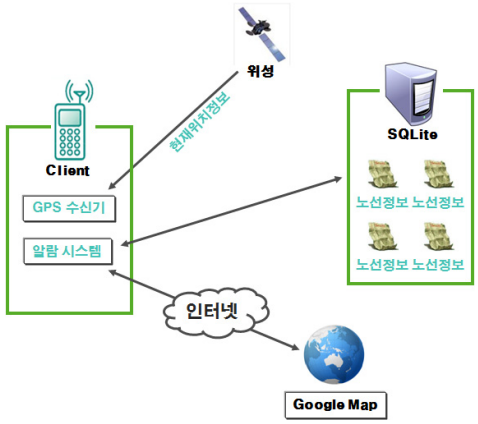
2.3. 개발환경

본 논문의 애플리케이션은 윈도우 XP 와 윈도우 7 환경에서 개발되었다. Java SE Development Kit (JDK6) 설치, Eclipse IDE 설치, Eclipse IDE 에 Android Plug-in 설치, Android SDK 2.1설치 후 실제 단말기와 에뮬레이터에서 개발되었다. 개발에 사용한 단말기는 구글에서 직접 설계해서 일명 '구글폰'이라고도 불리는 '넥서스원'이다. Wi-Fi, 멀티터치, 음성인식, GPS, MP3재생 등 본 애플리케이션이 작동하기에 충분한 기능을 갖추고 있어서 선택하였다[7].

3. 세부 설계 및 구현

3.1 시스템 구성

아래 [그림 2]는 버스알람 시스템의 구조를 보여준다.



(그림 2) 버스알람 시스템 구조

3.2 핵심 알고리즘

3.2.1 노선정보 다운로드

아래의 [그림 3]소스는 스트림 방식으로 파일의 끝을 만날 때 까지 지정해놓은 주소에 접속하여 노선데이터를 읽고, 미리 정해놓은 단말기의 저장 공간에 쓰기를 반복한다. 중간 중간에 'flagCancel' 값을 확인하여 취소 버튼을 눌렀는지 체크한 후 몇 퍼센트 까지 다운로드 되었는지 화면에 표시한다.

```
fileOutputStream = new FileOutputStream(file);
int cnt = 0;
while((cnt = inputStream.read(buf )) != -1
        && flagCancel == false) {
    fileOutputStream.write(buf,0,cnt);
    fileOutputStream.flush();
    nowDown += cnt;
    if(flagCancel == true)
        break;
    publishProgress( nowDown * 100 / totalDownSize );
```

(그림 3) 노선정보 다운로드 소스

3.2.2 현재위치 저장

[그림 4]는 데이터베이스에 레코드를 추가하기 위해 만든 함수이다. 파라미터로 이름과 위도 경도를 넘겨주면 그 정보를 데이터베이스에 추가한다. [그림 5]는 실제로 화면상에서 사용자에게 이름과 위도, 경도를 입력받고 이미 존재하는 이름인지, 한 글자이상 인지, 위도, 경도가 현재 잡히고 있는 상태인지를 체크한다. 모든 조건을 만족하면 [그림 4]의 함수로 추가할 데이터를 파라미터로 넘겨주면 쉽게 데이터베이스에 레코드를 추가할 수 있다.

```
public long createLocation(String name, int latitude,
        int longitude){
    ContentValues initialValues = new ContentValues();
    initialValues.put(KEY_NAME, name);
    initialValues.put(KEY_LATITUDE, latitude);
    initialValues.put(KEY_LONGITUDE, longitude);

    return mDb.insert(DATABASE_TABLE,
            null, initialValues);
}
```

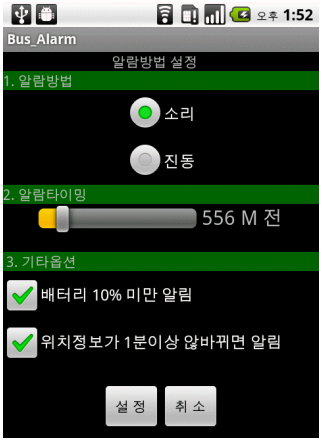
(그림 4) 데이터베이스에 레코드 추가하는 함수

```
if( isOverlapName(eName.getText().toString()) == true )
    Toast.makeText(SubActivity1.this,
            "같은 이름이 이미 존재합니다.",
            Toast.LENGTH_SHORT).show();
else{
    mUser_DB.createLocation(eName.getText().toString(),
            now_latitude, now_longitude);
    Toast.makeText(SubActivity1.this,"저장되었습니다.",
            Toast.LENGTH_SHORT).show();
    finish();
}
```

(그림 5) 현재위치저장 소스

3.2.3 알람설정

아래 [그림 6]은 알람방법 설정화면이다.



(그림 6) 알람방법설정 화면

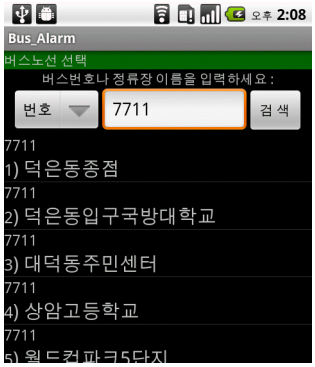
아래 [그림 7]과 같이 안드로이드에서는 “Shared Preferences”를 통해 Key와 간단한 데이터 값을 저장하고, 다른 Activity(화면) 에서 Key를 통해 그 값에 접근할 수 있다. 옵션메뉴에서는 사용자가 설정한 값을 Shared Preferences를 통해 저장하고, 다른 Activity(화면)에서 프로그램 중간에 옵션 값이 필요하면 미리 약속한 키를 통해 얻은 값으로 옵션에 따른 동작을 구현하였다.

```
// Shared Preference를 불러옵니다.
SharedPreferences option =
    getSharedPreferences("option",
            Activity.MODE_PRIVATE);
SharedPreferences.Editor editor =
    option.edit(); // Editor를 불러옵니다.

// 저장할 값들을 입력합니다.
editor.putBoolean("radio_sound",
        radio_sound.isChecked());
editor.putBoolean("radio_vibration",
        radio_vibration.isChecked());
editor.putInt("alarm_range",
        (alarm_range.getProgress() + 500) );
editor.putBoolean("battery_chk",
        op_batteryChk.isChecked());
editor.putBoolean("gps_chk", op_gpsChk.isChecked());

editor.commit(); // 저장합니다.
finish();
```

(그림 7) 알람방법설정 소스

3.2.4 버스노선 선택

[그림 8]은 버스노선 선택 실행화면이다. 사용자가 버스정류장의 이름이나 버스번호를 입력하면 관련된 정보를 데이터베이스에서 검색하여 검색창 아래에 리스트형태로 표시해준다. 사용자가 원하는 정류장을 선택하면 선택확인 후 목적지 알람시스템으로 해당 정류장의 위도, 경도, 이름을 넘기게 된다.



(그림 8) 버스노선선택 화면

[그림 9]의 함수는 파라미터로 넘겨받은 버스번호를 가지고 데이터베이스에 쿼리문을 날려서 그 결과를 리턴해 주는 함수이다. [그림 10]의 소스는 사용자에게 버스 번호나 정류장이름을 입력받고, 그 값을 [그림 9]의 함수에게 파라미터로 넘겨준다. 그리고 함수의 리턴 값을 가지고 Adapter를 생성한다. 이 Adapter 는 리스트 뷰와 연결되어 화면에 버스노선 리스트를 표시하게 된다.

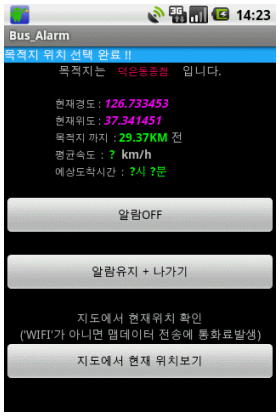
```
public Cursor fetchStationByNum(String busNum){
    Cursor mCursor =
        mDb.query(true, DATABASE_TABLE,
                new String[]{KEY_BUS_ID, KEY_BUS_NUM,
                KEY_STATION_NUM, KEY_STATION_NAME},
                KEY_BUS_NUM + "='" + busNum +"'",
                null, null, null, null, null);
    if(mCursor != null)
        mCursor.moveToFirst();
    return mCursor;
}
```

(그림 9) 버스번호로 데이터베이스 검색하는 함수

```
// DB에 검색 질의 요청
if((spnSearchOp.getSelectedItem()).equals("번호"))
    resCursor = mBus_DB.fetchStationByNum(str);
else
    resCursor = mBus_DB.fetchStationByName(str);

if(resCursor.getCount() > 0){
    // 리스트에 결과 출력
    String[] from =
        new String[]{Bus_DB.KEY_BUS_NUM,
            Bus_DB.KEY_STATION_NUM,
            Bus_DB.KEY_STATION_NAME};
    int[] to =
        new int[]{R.id.busdb_busNum,
            R.id.busdb_stationNum,
            R.id.busdb_stationName};
    SimpleCursorAdapter cursorAdapter =
        new SimpleCursorAdapter(this,
                R.layout.bus_db_list_in,
                resCursor, from, to);
    ListView listBusDB =
        (ListView)findViewById(R.id.busStation_list);
    listBusDB.setAdapter(cursorAdapter);
}
```

(그림 10) 버스노선선택 소스

3.2.5 목적지 알람



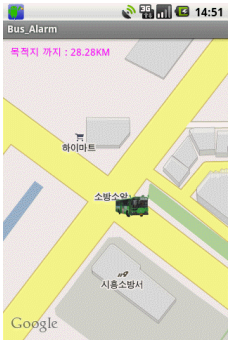
(그림 11) 목적지알람 화면

위 [그림11]은 목적지 알람화면으로써 이 프로그램의 가장 핵심적인 부분이다. 아래의 [그림 12]소스는 출발지, 목적지의 위도경도를 가지고 두 지점의 거리를 계산하는 소스이다. 지속적으로 두 지점의 거리를 갱신하다 설정한 알람반경 보다 짧은 거리면 “destination_alarm()” 함수를 호출한다. 이 함수는 사용자가 옵션에서 설정한 대로 소리 또는 진동을 발생시킨다.

```
double a_latitude = location.getLatitude(),
    a_longitude = location.getLongitude();
// 목적지까지 남은 거리 구하기
Location.distanceBetween(a_latitude, a_longitude,
        dest_latitude, dest_longitude, betweenAB);
if(betweenAB[0] <= alarm_range){
    // 알람 작동
    Toast.makeText(this,"알람범위 안에 들어왔습니다.",
            Toast.LENGTH_LONG).show();
    // 리스너 정지
    LocationManager locationM =
        (LocationManager)getSystemService
        (Context.LOCATION_SERVICE);
    locationM.removeUpdates(locationListener);
    // 옵션 설정대로 사용자에게 알림
    destination_alarm();
```

(그림 12) 목적지알람 소스

3.2.6 지도보기



(그림 13) 지도보기 화면

위의 [그림 13]은 구글 맵 위에 지속적으로 목적지와 현재위치를 표시하는 지도보기 화면이다. 아래의 [그림 14]소스는 목적지에 미리 정해놓은 목적지 이미지를 띄우고 현재위치에는 핸드폰의 이동에 따라 계속 갱신하면서 버스 이미지를 표시하는 소스이다.

```
// 목적지 위치 표시
MapView.LayoutParams mapMarkerParams = new
MapView.LayoutParams(LayoutParams.WRAP_CONTENT,
        LayoutParams.WRAP_CONTENT,
        dest_point, MapView.LayoutParams.TOP_LEFT);
ImageView mapMarker =
    new ImageView(getApplicationContext());
mapMarker.setImageResource(R.drawable.station);
map.addView(mapMarker, mapMarkerParams);

// 현재위치 표시
mapMarkerParams = new
MapView.LayoutParams(LayoutParams.WRAP_CONTENT,
        LayoutParams.WRAP_CONTENT,
        newPoint, MapView.LayoutParams.TOP_LEFT);
mapMarker = new ImageView(getApplicationContext());
mapMarker.setImageResource(R.drawable.bus);
map.addView(mapMarker, mapMarkerParams);
```

(그림 14) 지도보기 소스

4. 결론 및 향후 연구과제

본 애플리케이션을 사용하면 버스에서 잠이 들어도 목적지에 도착하면 일어날 수 있다. 부가적으로 목적지까지 대략적인 거리를 확인하거나 평균 속도를 통해 대략적인 도착시간도 알 수 있다. 또한 옵션을 통해 핸드폰의 전원이 10% 미만이거나 1분 동안 현재위치가 바뀌지 않으면 알려주는 옵션도 설정할 수 있다.

현재 서울특별시 교통정보센터 사이트[8]에서 제공하는 버스정류장 위치를 사용해 DB를 생성하였다. 하지만 서울지역 이외에는 버스정류장 위치를 구할 수 있는 곳이 없었다. 그래서 서울을 제외한 나머지 버스정류장들은 현재위치 저장 메뉴를 통해 사용자가 위치를 직접 저장해야 한다. 추후 다른 지역의 버스정류장 위치정보 업데이트가 필요하다. 또한 그래픽적인 부분을 좀 더 깔끔하게 표현하고, 옵션에서 알람소리를 벨소리 폴더에서 가져 오는 등 유저 인터페이스를 좀 더 개선해야할 필요가 있다.

참고문헌

[1] 스마트폰 시장현황 “http://www.mymits.net/496311”.
[2] 스마트폰 트래픽 “http://v.daum.net/link/8981057”.
[3] wakeupnow 소개 “http://blog.naver.com/lmg2738”.
[4] GPS알람 소개 “http://knmtskr.blog.me/150091042636”.
[5] 안드로이드 소개 “http://developer.android.com”.
[6] 안드로이드 게시판 “http://www.androidside.com”.
[7] 셰인콘더, 로런다시 지음(류광 역), 시작하세요! 안드로이드 프로그래밍: 모바일 소프트웨어 개발, 위키북스, 2009년 11월.
[8] 서울특별시 교통정보센터 “http://topis.seoul.go.kr”.