

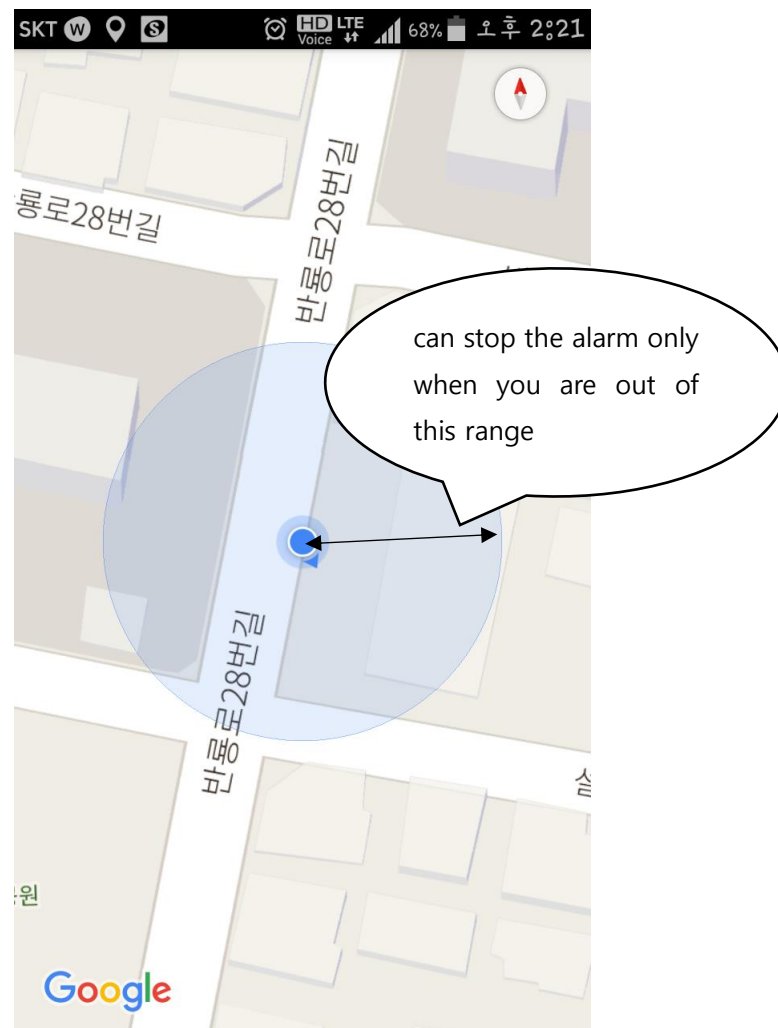
## - Android Alarm Application

### 1. Primary function

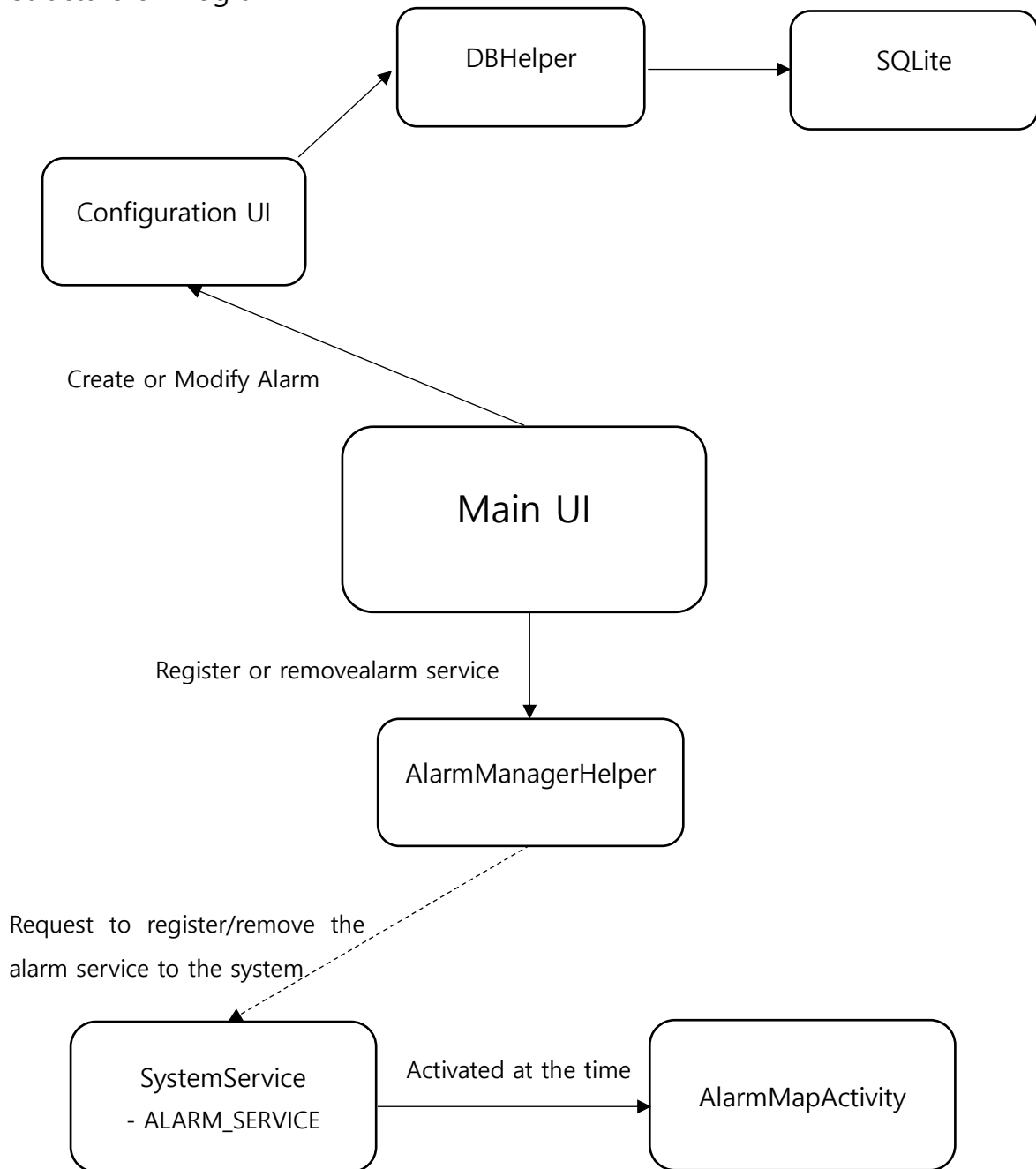
- Alarm when the user set to wake up an Olympian-Sleeper

### 2. Purpose

- For me who have heavy sleep in the morning
- increase the probability to wake up by moving to stop the alarm



### 3. Structure of Program



#### 4. The way of implementation

- Get alarm service at the set time

Register to system service

```
public static void setAlarm(Context context, Calendar calendar, PendingIntent pIntent){  
    AlarmManager alarmManager = (AlarmManager) context.getSystemService(Context.ALARM_SERVICE);  
    alarmManager.set(AlarmManager.RTC_WAKEUP, calendar.getTimeInMillis(), pIntent);  
}
```

- Save the alarm service information to Embedded SQLite

```
public class AlarmDBHelper extends SQLiteOpenHelper implements BaseColumns{  
    public static final int DATABASE_VERSION = 1;  
    public static final String DATABASE_NAME = "alarm5.db";  
    public static final String SQL_TABLE_NAME = "alarm";  
    public static final String DELIMITER = "|";  
  
    public AlarmDBHelper(Context context) {  
        super(context, DATABASE_NAME, null, DATABASE_VERSION);  
        // TODO Auto-generated constructor stub  
    }  
  
    public long createAlarmClass(AlarmClass obj){  
        ContentValues values = new ContentValues();  
        values.put(COLUMN_NAME_ALARM_TIME_HOUR, obj.timeHour);  
        values.put(COLUMN_NAME_ALARM_TIME_MINUTE, obj.timeMinute);  
        values.put(COLUMN_NAME_ALARM_REPEAT_WEEKLY, obj.isRepeat);  
        values.put(COLUMN_NAME_ALARM_TONE, obj.alarmTone != null ? obj.alarmTone.toString() : "");  
        values.put(COLUMN_NAME_ALARM_ENABLED, obj.isEnabled);  
        values.put(COLUMN_NAME_ALARM_VOLUME, obj.volume);  
        values.put(COLUMN_NAME_ALARM_RADIUS, obj.radius);  
  
        String repeatingDays = "";  
        for (int i = 0; i < 7; ++i) {  
            repeatingDays += obj.getRepeatingDay(i) + DELIMITER;  
        }  
        values.put(COLUMN_NAME_ALARM_REPEAT_DAYS, repeatingDays);  
  
        return getWritableDatabase().insert(TABLE_NAME, null, values);  
    }  
}
```

```

public List<AlarmClass> getAlarms(){
    SQLiteDatabase db = this.getReadableDatabase();

    String query = "SELECT * FROM " + TABLE_NAME;

    Cursor c = db.rawQuery(query, null);

    List<AlarmClass> alarmList = new ArrayList<AlarmClass>();

    while(c.moveToNext())
        alarmList.add(getAlarmClass(c));

    if(alarmList.isEmpty())
        return null;
    else
        return alarmList;
}

public int removeAlarm(long id){
    return getWritableDatabase().delete(TABLE_NAME, _ID + " = ?", new String[] { String.valueOf(id) });
}

```

<DB Insert, Update, Remove, and Get>

- Using google map & setting the condition to be able to stop the alarm

1. Setting an API key for google map

```

<meta-data
    android:name="com.google.android.maps.v2.API_KEY"
    android:value="AIzaSyCFVQkAfwrQLGSN17L70cV1HK55e7Euth8" />
<meta-data android:name="com.google.android.gms.version"
    android:value="@integer/google_play_services_version"/>

```

## 2. interface for using Google Map

```
public class AlarmMapActivity extends Activity
    implements GooglePlayServicesClient.ConnectionCallbacks,
    GooglePlayServicesClient.OnConnectionFailedListener,
    LocationListener{

    private LocationClient mLocationClient;
    private WakeLock mWakeLock;
    private MediaPlayer mPlayer;
    private AudioManager mAm;
    GoogleMap map;

    private final String TAG = this.getClass().getSimpleName();
    private String PROXIMITY_ALERT = "com.example.embedded_text.PROXIMITY_ALERT";
    private final static int CONNECTION_FAILURE_RESOLUTION_REQUEST = 9000;
    private final static int WAKELOCK_DELAY = 1000 * 60 * 30; // 30 min

    boolean isOutside;
    ProximityAlertReceiver alertReceiver;
```

## 3. Set current position and stop condition

```
alertReceiver = new ProximityAlertReceiver();
IntentFilter filter = new IntentFilter(PROXIMITY_ALERT);
registerReceiver(alertReceiver, filter);
```

<Register the Proximity Alert Receiver>

```
private void addProximityAlert(LatLng ll, int radius){
    LocationManager locMng =
        (LocationManager) getSystemService(LOCATION_SERVICE);
    String bestProvider = getBestProvider(locMng);
    if(bestProvider == null)
        bestProvider = LocationManager.GPS_PROVIDER;
    locMng.requestLocationUpdates(bestProvider, 500, 0.5f, this);

    Intent intent = new Intent(PROXIMITY_ALERT);
    PendingIntent pendingIntent =
        PendingIntent.getBroadcast(this, -1, intent, PendingIntent.FLAG_UPDATE_CURRENT);

    locMng.addProximityAlert(ll.latitude, ll.longitude, radius, -1, pendingIntent);
}
```

<Register the data to Proximity Alert with the radius the user wants>

```

class ProximityAlertReceiver extends BroadcastReceiver{
    String key = LocationManager.KEY_PROXIMITY_ENTERING;

    @Override
    public void onReceive(Context context, Intent intent) {
        // TODO Auto-generated method stub
        if(intent.getBooleanExtra(key, false)){
            isOutside = false;
        } else {
            isOutside = true;
            Toast.makeText(getBaseContext(), "Well done!!",
                Toast.LENGTH_SHORT).show();
        }
    }
}

```

<Implementation of receiver>

- Play the tone

```

String strToneUri = getIntent().getStringExtra(AlarmManagerHelper.TONE);
startMedia(strToneUri);

```

```

private void startMedia(String strUri){
    mPlayer = new MediaPlayer();
    try{
        if(strUri != null &&
            !strUri.equals("")){
            Uri toneUri = Uri.parse(strUri);
            if(toneUri != null){
                mPlayer.setDataSource(this, toneUri);
                mPlayer.setAudioStreamType(AudioManager.STREAM_ALARM);
                mPlayer.setLooping(true);
                mPlayer.prepare();
                mPlayer.start();
            }
        }
    } catch(Exception e){
        e.printStackTrace();
    }
}

```

---

- make the back button useless when the alarm service is activated

```
@Override  
public void onBackPressed(){  
    /* do nothing when back button pressed */  
}
```

(★ A few months ago, the ransomware tortured me. So, I lost most of the things related past works. - I have no real face of App.)