



국민대학교
소프트웨어융합대학
소프트웨어학부

캡스톤 디자인 I

종합설계 프로젝트

프로젝트 명	Smarlarm(스말람)
팀 명	<i>Sweetleep</i>
문서 제목	중간보고서(2 차)

Version	1.4
Date	2020-MAY-28

팀원	홍성표
	고민욱
	박대영
	박영민
	서민주
	양경식

 국민대학교 소프트웨어학부 캡스톤 디자인 I	중간보고서(2차)		
	프로젝트 명	Smarlarm(스말람)	
	팀 명	Sweetleep	
	Confidential Restricted	Version 1.4	2020-05-28

문서 정보 / 수정 내역

CONFIDENTIALITY/SECURITY WARNING	
<p>이 문서에 포함되어 있는 정보는 국민대학교 소프트웨어융합대학 소프트웨어학부 개설 교과목 캡스톤 디자인 수강 학생 중 프로젝트 " Smarlarm(스말람)"를 수행하는 팀 "SWEETLEEP"의 팀원들의 자산입니다. 국민대학교 소프트웨어학부 및 팀 "SWEETLEEP"의 팀원들의 서면 허락없이 사용되거나, 재가공 될 수 없습니다.</p>	

Filename	15조_2차중간보고서_V1.4
원안작성자	홍성표
수정작업자	박영민, 박대영, 고민욱, 서민주

수정날짜	대표수정자	Revision	추가/수정 항목	내 용
2020-05-27	박영민	1.0	수면패턴관련 추가 및 수정	프로젝트목표(수면패턴), 연구내용(수면패턴), 수행내용(수면패턴), 참고문헌
2020-05-28	박대영	1.1	서버내용 추가	
2020-05-28	박영민	1.2	수면패턴 내용수정	
2020-05-28	고민욱	1.3	역알람 내용 추가	
2020-05-28	서민주	1.4	알람 내용 추가	프로젝트목표(스마트알람)

 국민대학교 소프트웨어학부 캡스톤 디자인 I	중간보고서(2차)		
	프로젝트 명	Smarlarm(스말람)	
	팀 명	Sweetleap	
	Confidential Restricted	Version 1.4	2020-05-28

내용

1	프로젝트 목표	4
2	수행 내용 및 중간결과	5
2.1	계획서 상의 연구내용	5
2.1.1	수면패턴 측정 알고리즘	5
2.1.2	알람 추천 서비스 알고리즘	7
2.1.3	프론트엔드	9
2.2	수행내용	13
2.2.1	수면패턴 측정 알고리즘	13
2.2.2	알람 추천 서비스	17
2.2.3	프론트엔드	19
3.	수정된 연구 내용 및 수정사항	25
3.1	수정된 내용	25
3.1.1	기존 시스템과 차별점	25
4.	향후 추진 내용	26
4.1	GPS연동 및 경로 파악	26
4.2	날씨 연동	27
5.	참고문헌	28

 국민대학교 소프트웨어학부 캡스톤 디자인 I	중간보고서(2차)		
	프로젝트 명	Smarlarm(스말람)	
	팀 명	Sweetleep	
	Confidential Restricted	Version 1.4	2020-05-28

1 프로젝트 목표

본 프로젝트는 현대인들이 자신의 생활 패턴과 수면 패턴을 파악하고 그에 맞게 일정을 관리할 수 있도록 도와주고자 합니다. 또한 현대인들에게 필수가 된 알람 서비스는 실시간적으로 변화하는 교통상황 및 날씨를 파악하여 가장 최적의 알람 시간을 추천해 주는 데에 목표가 있습니다. 다음과 같은 3 가지 서비스를 중점적으로 제공합니다..

- 수면 패턴 분석

스마트폰에 내장된 가속도계를 이용하여 자신의 수면 중 움직임을 측정합니다. 또한 Google fit 사용자라면 심박수 데이터를 얻어올 수 있습니다. 측정된 가속도계 데이터와 심박수 데이터를 조합하여, 수면시간 동안의 수면단계(깊은 잠, 얇은 잠)를 결정합니다.

- 스마트 알람 서비스

현대인에게 알람은 필수적인 기능이 되었습니다. 이에 따라 기본적인 알람을 넘어 스마트 알람 기능을 제공합니다. Google 캘린더와 연동하여 아침에 일어나 준비하는 시간과 교통수단을 입력 받으면 적당한 알람 시간을 추천해줍니다. 교통 상황이 좋지 않다면 조금 일찍 알람을 울립니다. 수면 분석 결과를 토대로 수면시간이 부족하다 판단되면 역으로 잠자리에 들기를 추천하는 알람 서비스를 제공합니다.

- 정보제공

정보제공은 한 달 혹은 일주일 단위로 자신이 과거의 수면 패턴 그래프를 모아서 한 번에 보여주는 기능입니다. 이런 기능을 통하여 자신이 언제 잠을 깊이 잤는지를 알 수 있으며 혹은 뒤척임, 소리를 통해 사용자의 건강 상태 또한 체크할 수 있습니다.

2 수행 내용 및 중간결과

2.1 계획서 상의 연구내용

2.1.1 수면패턴 측정 알고리즘

기존 수면패턴 분석 알고리즘은 가속도계 센서를 이용하여 각 수면단계를 결정하는 방식이었지만, 스마트폰의 가속도계 측정방식의 여러가지 외부요인들에 의한 제한사항이 많다는 피드백을 수용하여 좀 더 정확한 방식인 심박수 데이터를 이용해서 각 수면단계를 결정해보기로 하였습니다.

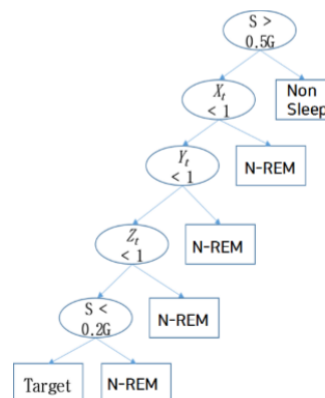
결국 심박수 데이터를 얻어오려면 Smart watch와 같은 웨어러블 기기를 활용해야 하는데, 시중에 나와있는 웨어러블 기기에서 직접적으로 심박수 데이터를 받아올 수 있는 오픈 된 API를 찾는데 어려움을 겪었습니다. 가능한 방법을 검색해본 결과 다양한 웨어러블 기기(샤오미, 핏빗 등)와 연동이 가능한 Google fit (<https://developers.google.com/fit>)이라는 플랫폼을 사용해보기로 하였습니다.

본 플랫폼은 다양한 브랜드의 웨어러블 기기와 연동이 가능하며, 연동 작업 후에는 웨어러블 기기에서 얻어온 다양한 건강정보 (심박수, 걸음수 등)들이 각 개인별 구글계정에 저장되며, API를 이용하여 조회할 수 있습니다. 얻어온 심박수를 통해, 수면단계를 결정하는 알고리즘은 다음 논문을 참조하여서 진행하였습니다.

심박수, 가속도 센서를 활용한 수면 유도 시스템

김주연*, 김정은, 이승형, 이현수
경희대학교 컴퓨터공학과

Time	HR	X	Y	Z	X_t	Y_t	Z_t	G	S
15:56	87	23	43	130	-	-	-	24	17
15:57	83	15	42	126	8	1	4	24	13
15:58	83	60	35	124	45	7	2	24	13
15:59	82	78	35	126	18	0	2	24	12



[그림 2] 수면 상태 판정을 위한 의사결정 트리 예시

 국민대학교 소프트웨어학부 캡스톤 디자인 I	중간보고서(2차)		
	프로젝트 명	Smarlarm(스말람)	
	팀 명	Sweetleap	
	Confidential Restricted	Version 1.4	2020-05-28

위의 왼쪽 그림에서

G는 (정상시 평균심박수 - 휴식시 최저심박수) 를 의미하며,

S는 (현재심박수 - 휴식시 최저심박수) 를 의미합니다.

X,Y,Z 는 가속도계에서 얻어온 각 축 데이터를 의미합니다.

각 수면단계(깨어남, 깊은 잠, 얇은 잠)를 구분 짓는 의사 결정 트리(우측방향이 YES)는 위의 오른쪽 그림과 같습니다. 먼저 현재 심박수를 이용해 휴식 기 심박수 여부를 확인한 후, 사용자의 움직임 여부를 확인합니다. 그 후, 다시 심박수를 이용해 깊은 잠 여부를 확인합니다.

현재 이 논문에서 수면단계를 구분 짓는 센서데이터는 가속도계, 심박수 2가지로 3단계에 걸쳐서 수면단계를 구분하고 있습니다. 본 프로젝트에서는 심박수는 Google fit을 이용해 비교적 정확한 값을 얻을 수 있지만, 사용자의 움직임 여부를 확인하는 부분은 스마트폰의 가속도계를 이용한 측정값에 의존해야 하기 때문에 그 정확도가 상기 논문보다는 떨어질 수도 있다고 생각합니다.

여러가지 제약사항에 걸리는 상황이지만, 데이터를 얻어와서 분석을 해본다는 것에 의미를 두고 진행해보고자 합니다. 어플리케이션의 시작부분에 사용자에게 자신의 사용환경을 input으로 받아서 다음과 같은 2가지 시나리오로 수면단계분석 결과를 제공하고자 합니다.

- 연동된 **Google fit 계정이 있는 경우** : 스마트폰 가속도계 + 심박수 데이터
- **없는 경우** : 스마트폰 가속도계

사용자가 침대에 누워서 측정시작 버튼을 누르더라도 바로 잠이 들었다고 판단할 수 없기 때문에 다음과 같은 2가지 조건일 때 수면상태라고 판단할 예정입니다.

1. 심박수가 휴식 기 이하
2. 최소 15분 동안 움직임이 없을 때 (움직임 값이 일정 값 이하일 때)

또한 수면상태에서의 깊은 잠, 얇은 잠 여부는 다음과 같은 조건을 걸 예정입니다.

1. IF ($S < 0.2G$) -> 깊은 잠
2. ELSE -> 얇은 잠

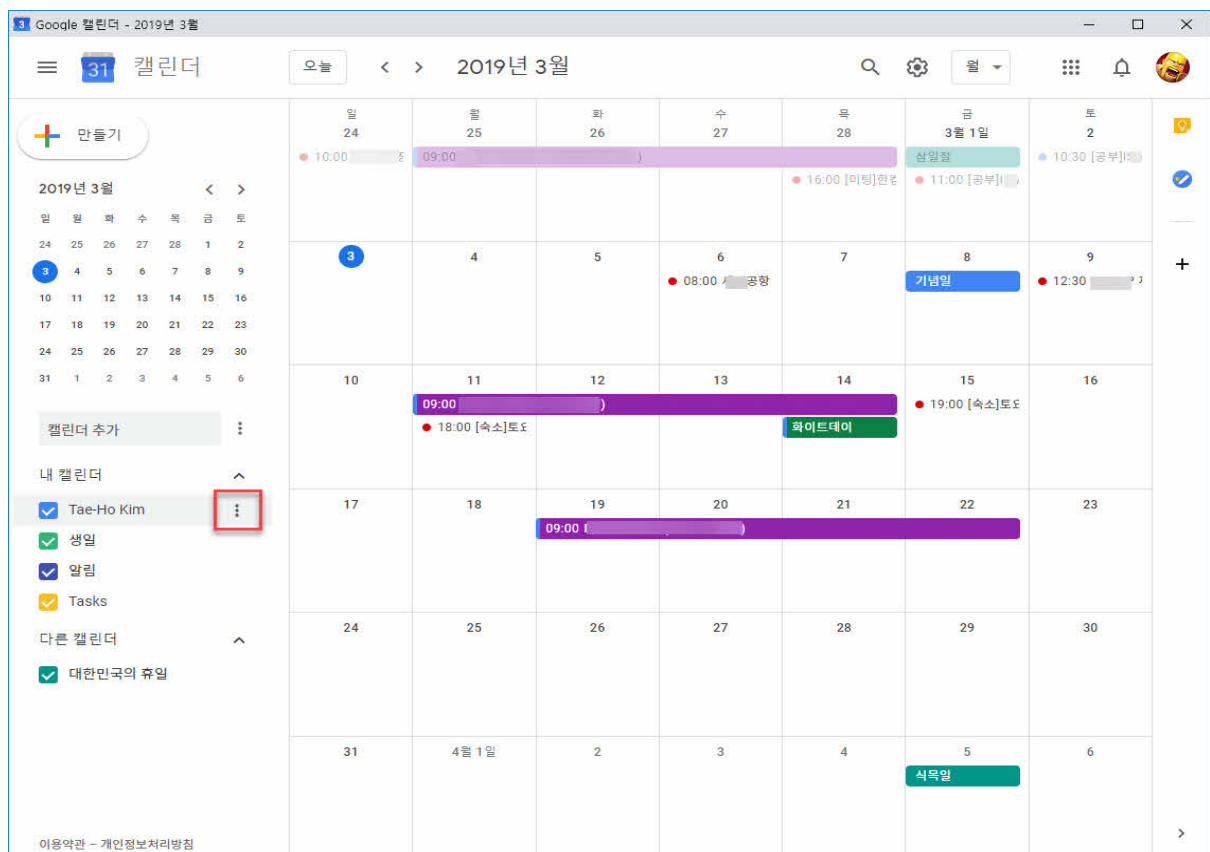
 국민대학교 소프트웨어학부 캡스톤 디자인 I	중간보고서(2차)		
	프로젝트 명	Smarlarm(스말람)	
	팀 명	Sweetleep	
	Confidential Restricted	Version 1.4	2020-05-28

2.1.2 캘린더 기반 알람 생성 서비스

바쁜 현대인들에게 시간은 금과 같다. 어떻게 하면 더욱 시간을 효율적으로 사용할 것인지에 대한 고민은 끝없이 진행중에 있습니다. 요즘 중, 고등학교 학생들부터 대학생, 직장인들까지 일정이 많아지고 복잡해짐에 따라 일정관리가 필수가 되었고, 자연스럽게 일정관리 어플리케이션 사용자가 많아지고 있습니다. 대표적 일정관리 어플리케이션의 유형인 캘린더를 알람 어플리케이션과 연동시켜 사용자의 편의를 더욱 높이고자 다음과 같은 기능을 구현 할 예정입니다.

우리 어플에서는 스마트폰에 연동되어 있는 Google 계정정보를 이용해 Google Calendar 와 연동합니다. 그리고 사용자에게 준비시간/소요시간 및 주교통수단을 입력 받아 사용자가 언제 출발해야 할지 혹은 언제 일어나야 하는지 계산하여 제공합니다.

사용자는 제공된 시간을 기반으로 알람을 설정할지 선택할 수 있습니다.



이 기능의 핵심적인 부분은 일정 등록이다. Google Calendar API 를 통해 사용자가 기존에 등록해 놓은 일정을 불러오도록 하여 알람을 설정하는데 도움을 줍니다.

 국민대학교 소프트웨어학부 캡스톤 디자인 I	중간보고서(2차)		
	프로젝트 명	Smarlarm(스말람)	
	팀 명	Sweetleep	
	Confidential Restricted	Version 1.4	2020-05-28



또한 사용자가 준비시간, 목적지, 교통수단을 입력하게 되면 일정시작시간에 맞출 수 있게 알람 시간을 조정합니다. 만약 목적지와 교통수단이 입력된다면, 교통상황을 체크하고 교통체증이 있는 경우 계획보다 조금 더 일찍 일어날 수 있는 알람을 울립니다. 또는 알람 종료시간 기준으로 기상 상황(눈/비/태풍 등)이 좋지 않다면 역시 조금 더 일찍 알람을 울립니다.

이 기능 또한 사용자가 원하지 않을 경우 OFF할 수 있도록 합니다.

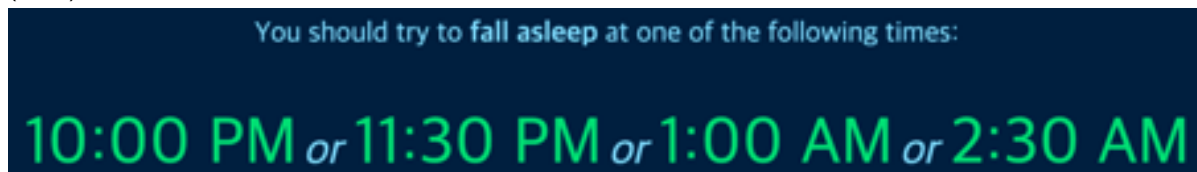
 국민대학교 소프트웨어학부 캡스톤 디자인 I	중간보고서(2차)		
	프로젝트 명	Smarlarm(스말람)	
	팀 명	Sweetleap	
	Confidential Restricted	Version 1.4	2020-05-28

2.1.3 역알람 기능

기존의 알람 어플들이 기상시간만 추천해주는 것과 달리 잠들어야 할 시간을 추천해주는 기능을 제공합니다.

사용자의 캘린더를 바탕으로 알람 시간을 추천 받으면 그 시간에 맞추어 사용자가 잠들면 좋은 시간을 푸시알림으로 제공합니다.

만약 사용자가 7:00에 일어나야 하는 경우
(예시)



위와 같은 시간들을 사용자에게 알려주고 알맞은 시간이 있으면 사용자가 선택하게 합니다.

위 시간들은 다음과 같이 계산됩니다. 먼저 비REM - REM의 주기인 90분을 1 cycle로 하여 추천된 기상시간으로 부터 약 4~6 cycle 전의 시간을 잠들어야 할 시간으로 계산합니다. 그 시간에 사용자가 침대에 누워서 잠들기까지의 시간을 추가합니다. 이 시간은 사람마다 많은 차이를 보이므로 기본 15분으로 설정해둔 뒤 수면 분석을 통해 나타난 사용자의 잠들기까지 걸리는 시간을 저장하고 기존의 시간에 더해 평균 시간을 구합니다. 따라서 사용자가 어플을 사용하는 기간이 길수록 사용자에게 더욱 최적화된 시간을 제공하리라 예상됩니다.

2.1.4 프론트엔드

프론트엔드에서는 사용자가 수면패턴분석 기능 사용에 어려움을 느끼지 않도록 어플리케이션 첫 실행 시 사용방법을 알려주는 화면을 구성하여 어려움을 줄이고자 합니다. 또한 최대한 직관적이고 복잡하지 않게 세가지 탭들로 구성된 Bottom Navigation Bar로 Fragment 간의 이동을 지원합니다.

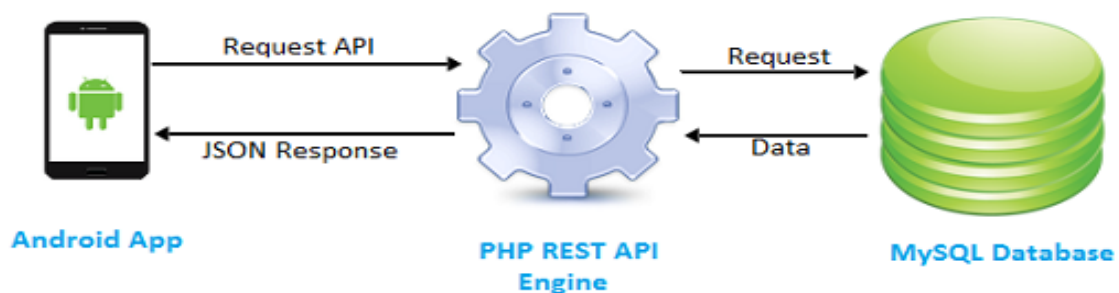
알람 목록은 등록된 알람 갯수가 많은 경우를 대비해 Recyclerview를 이용해 view를 재사용하도록 했습니다.

 국민대학교 소프트웨어학부 캡스톤 디자인 I	중간보고서(2차)		
	프로젝트 명	Smarlarm(스말람)	
	팀 명	Sweetleep	
	Confidential Restricted	Version 1.4	2020-05-28

2.1.5 데이터베이스 & 서버

1) 데이터베이스

데이터베이스는 사용자들의 캘린더를 저장하는 용도로 사용하기 위함입니다. MySQL를 사용하여 데이터베이스를 만듭니다. 안드로이드에서는 주로 보안상 이유로 직접 접근하지 않고 다른 API를 거쳐서 접근합니다.



안드로이드에서는 데이터베이스를 구현할 수 있도록 SQLiteDatabase 라는 DATABASE 를 제공해 줍니다. 만약 데이터를 저장해서 상요하게 되는 경우가 생길 경우에는 DatabaseHelper 를 사용하여 좀더 편리하게 DATABASE 를 관리 할 수 있습니다. DB 를 사용하기 위해서는 먼저 DB 를 생성하합니다. DB 는 테이블 구조로 데이터를 관리하고 있으므로 TABLE 구조를 만들어서 사용자의 캘린더를 저장합니다.

```
// DataBase Table
```

```
public final class DataBases {
```

```
    public static final class CreateDB implements BaseColumns{
```

```
        public static final String NAME = "name";
```

```
        public static final String CONTACT = "schedule";
```

```
        public static final String EMAIL = "email";
```

```
        public static final String _CREATE =
```

```
            "create table "+_TABLENAME+"("
```

```
                +_ID+" integer primary key autoincrement, "
```

```
                +NAME+" text not null , "
```

 국민대학교 소프트웨어학부 캡스톤 디자인 I	중간보고서(2차)		
	프로젝트 명	Smarlarm(스말람)	
	팀 명	Sweetleap	
	Confidential Restricted	Version 1.4	2020-05-28

```

+CONTACT+" text not null , "
+EMAIL+" text not null );";
}
}

```

더불어 DB관리를 통하여 데이터베이스를 최초 DB를 만들 때 한번만 호출하고 버전이 업데이트 되었을 경우 DB를 다시 만들어주는 방식을 사용합니다.

```

public class DbOpenHelper {

    private static final String DATABASE_NAME = "addressbook.db";
    private static final int DATABASE_VERSION = 1;
    public static SQLiteDatabase mDB;
    private DatabaseHelper mDBHelper;
    private Context mContext;

    private class DatabaseHelper extends SQLiteOpenHelper{
        // 생성자
        public DatabaseHelper(Context context, String name,
            CursorFactory factory, int version) {
            super(context, name, factory, version);
        }
        // 최초 DB를 만들때 한번만 호출된다.
        @Override
        public void onCreate(SQLiteDatabase db) {
            db.execSQL(DataBases.CreateDB_CREATE);
        }
        // 버전이 업데이트 되었을 경우 DB를 다시 만들어 준다.
        @Override
        public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
            db.execSQL("DROP TABLE IF EXISTS "+DataBases.CreateDB_TABLENAME);
            onCreate(db);
        }
    }
}

```

 국민대학교 소프트웨어학부 캡스톤 디자인 I	중간보고서(2차)		
	프로젝트 명	Smarlarm(스말람)	
	팀 명	Sweetleap	
	Confidential Restricted	Version 1.4	2020-05-28

2) 서버

사용자가 알람을 설정하면 알람 1 시간전부터 어플에서 5 분단위로 request 를 보낸다. 요청시 데이터의 타입은 content-value 이다. 서버에서는 요청받은 request 를 처리하여 원래의 기상시간보다 일찍 일어나야한다던가 원래 기상시간으로 일어나도 된다는 정보를 처리하여 어플로 return 해준다.



 <div> 국민대학교 소프트웨어학부 캡스톤 디자인 I </div>	중간보고서(2차)		
	프로젝트 명	Smarlarm(스말람)	
	팀 명	Sweetleap	
	Confidential Restricted	Version 1.4	2020-05-28

2.2 수행내용

2.1.1 수면패턴 측정 알고리즘

* 스마트폰 가속도계 측정

스마트폰에서 가속도계를 이용하여 움직임을 측정하는 코드는 그대로 활용할 예정입니다. 하지만 Google fit에서 얻어온 심박수 데이터는 1분단위로 오기 때문에, 기존의 15분단위에서 1분단위의 움직임을 담는 방식으로 코드를 변경하였습니다.

움직임이 감지된 경우 다음과 같은 onSensorChanged() 메소드에서 측정된 가속도계 좌표를 기반으로 속도(speed)를 구한 후 그 차이를 측정합니다. 그 속도가 threshold(민감도) 보다 높다면, 의미 있는 움직임이라 보고 setMotioncounter() 메소드를 통해 해당 구간의 움직임 횟수를 증가시켜줍니다. 1분 단위로 움직임을 측정해서 저장합니다.

```

@Override
public void onSensorChanged(SensorEvent event) {
    if (event.sensor.getType() == Sensor.TYPE_ACCELEROMETER) {
        // 움직인 속도를 구하는 과정
        long currentTime = System.currentTimeMillis();
        long gabOfTime = (currentTime - lastTime);
        if (gabOfTime > 100) {
            lastTime = currentTime;
            x = event.values[SensorManager.DATA_X];
            y = event.values[SensorManager.DATA_Y];
            z = event.values[SensorManager.DATA_Z];
            speed = Math.abs(x + y + z - lastX - lastY - lastZ) / gabOfTime * 10000;

            if ( (speed > SHAKE_THRESHOLD) ){

                Calendar cal = Calendar.getInstance();
                long t = cal.getTimeInMillis() / (60*1000);
                int time = (int) t; // 현재시각(분)

                // 해쉬맵에 저장
                this.setMotionCounter(time);
            }
        }
    }
}

```



```
}  
lastX = event.values[DATA_X];  
lastY = event.values[DATA_Y];  
lastZ = event.values[DATA_Z];  
}  
}
```

*Google fit 에서 데이터 가져오기

피트니스 객체에서 가져올 데이터(심박수) 설정 및 권한설정 처리

```
// 피트니스 객체 지정  
fitnessOptions = FitnessOptions.builder()  
    .addDataType(DataType.TYPE_HEART_RATE_BPM, fitnessOptions.ACCESS_READ)  
    .addDataType(DataType.TYPE_HEART_RATE_BPM, fitnessOptions.ACCESS_WRITE)  
    .build();  
  
// Google SignIn  
GoogleSignInAccount account = GoogleSignIn.getAccountForExtension(this, fitnessOptions);  
  
// Google Account check  
if (!GoogleSignIn.hasPermissions(account, fitnessOptions)) {  
    GoogleSignIn.requestPermissions(  
        this, // your activity  
        GOOGLE_FIT_PERMISSIONS_REQUEST_CODE, // e.g. 1  
        account,  
        fitnessOptions);  
} else {  
    readRecentHeartLate(); // 최근 7 일동안 평균심박수, 최저심박수 구하는 method  
}
```

권한이 다 ok라면 최근 7일동안의 평균 심박수 및 최저 심박수를 결정. **평균 심박수**는 휴식기 심박수(100이하) 들의 평균을 구하고, **최저 심박수**는 심박수 데이터들이 가끔 비정상적으로 낮게 측정될 수도 있으므로 휴식기 심박수 중 하위 5%의 평균값으로 설정할 예정입니다.

```
// 휴식기 심박수의 데이터 갯수 및 평균 구하기  
for (String s : heartLateList) {  
    // 100 미만일때 (움직임이 없을 때 심박수범위 60 < 100)  
    if (Float.valueOf(s) < 100) {  
        rest_count++;  
    }  
}
```



```

        rest_total += Float.valueOf(s);
    }
}
if(rest_count != 0) {
    rest_avg = rest_total / rest_count; // 휴식기 심박수 평균
}

// 휴식기 심박수의 하위 5 퍼센트 평균 구하기
rest_min_count = (int) (rest_count * 0.05); // 휴식기 심박수 중 하위 5 퍼센트 index 값

// 오름차순으로 정렬하기 위해 새로운 arraylist 에 index 값까지 담은 후 정렬
for(int i=0; i<heartLateList.size(); i++){
    minHeartLateList.add(Math.round(Float.valueOf(heartLateList.get(i))));
}
Collections.sort(minHeartLateList);

for (int i = 0; i < rest_min_count; i++) {
    rest_min_total += minHeartLateList.get(i);
}
if(rest_min_count != 0) {
    rest_min = rest_min_total / rest_min_count; // 휴식기 심박수의 하위 5 퍼센트 평균
}

```

startTime(수면측정시작버튼을 누른 시각), endTime(알람이 울린 시각) 사이의 각 개인별 심박수 데이터를 얻어옵니다. 원래 1분단위로 와야 정상이지만, 비는 값들이 생길 수도 있으므로 datapoint가 몇 개인지 체크 후 사이사이 비는 값들은 평균 값으로 처리할 예정입니다.

```

// 수면시간 동안의 심박수 data read
private void readSleepHeartLate(long startTime, long endTime) {

    // 심박수 데이터 getHistoryClient (2020 년 5 월 19 일 수면기록)
    Fitness.getHistoryClient(this,
        GoogleSignIn.getLastSignedInAccount(this))
        .readData(new DataReadRequest.Builder()
            .read(DataType.TYPE_HEART_RATE_BPM)// 심박수
            .setTimeRange(startTime, endTime, TimeUnit.MILLISECONDS)
            .build())

```

 국민대학교 소프트웨어학부 캡스톤 디자인 I	중간보고서(2차)		
	프로젝트 명	Smarlarm(스말람)	
	팀 명	Sweetleep	
	Confidential Restricted	Version 1.4	2020-05-28

```

.addOnSuccessListener(new OnSuccessListener<DataReadResponse>() {

    @Override
    public void onSuccess(DataReadResponse response) {
        DataSet dataSet = response.getDataSet(DataType.TYPE_HEART_RATE_BPM);
        // 수면시간 동안의 심박수 데이터 설정 (sleepHeartLateList)
        setSleepHeartLateValue(dataSet);
    }
});
}

```

하단은 각 개인별 심박수 통계화면과 5월 19일 수면시간 기반으로 한 심박수 데이터를 얻어온 후 같은 시간내의 심박수를 적용시켜서 각 수면단계를 구분해본 결과 화면입니다.

GoogleFit	GoogleFit
End time: 28일 00시 35분 Value: 93.0 Count of data: 9036.0, AVG: 89.33332536520585, MIN: 50.0	Count of datapoints: 19b 총 수면 시간: 389, 기상: 10, 알은수면: 27%, 깊은수면: 110
19일 수면기록 기반한 수면기록 보기	2b49b959: 1
Time: 16일 00시 01분, Value: 109.0	2b49b960: 1
Time: 16일 00시 02분, Value: 109.0	2b49b961: 1
Time: 16일 00시 03분, Value: 109.0	2b49b962: 1
Time: 16일 00시 04분, Value: 84.0	2b49b963: 1
Time: 16일 00시 05분, Value: 74.0	2b49b964: 1
Time: 16일 00시 06분, Value: 95.0	2b49b965: 1
Time: 16일 00시 07분, Value: 94.0	2b49b967: 1
Time: 16일 00시 08분, Value: 80.0	2b49b969: 1
Time: 16일 00시 09분, Value: 96.0	2b49b971: 1
Time: 16일 00시 10분, Value: 103.0	2b49b973: 1
Time: 16일 00시 11분, Value: 109.0	2b49b975: 1

수면시간 동안의 수면단계를 시간대별로 시각화 해서 볼 수 있는 부분은 현재 코드는 1분별로 수면단계를 나누기 때문에, 1분이상 최소 시간단위를 몇 분으로 결정하여 구간별로 필터링을 할지 고민중입니다. 그 최소 시간단위를 정하고 필터링이 된 데이터는 다음과 같은 가로 막대그래프 형식으로 색깔을 다르게 해서 얇은 수면, 깊은 수면 시간을 볼 수 있게 나타낼 예정입니다.

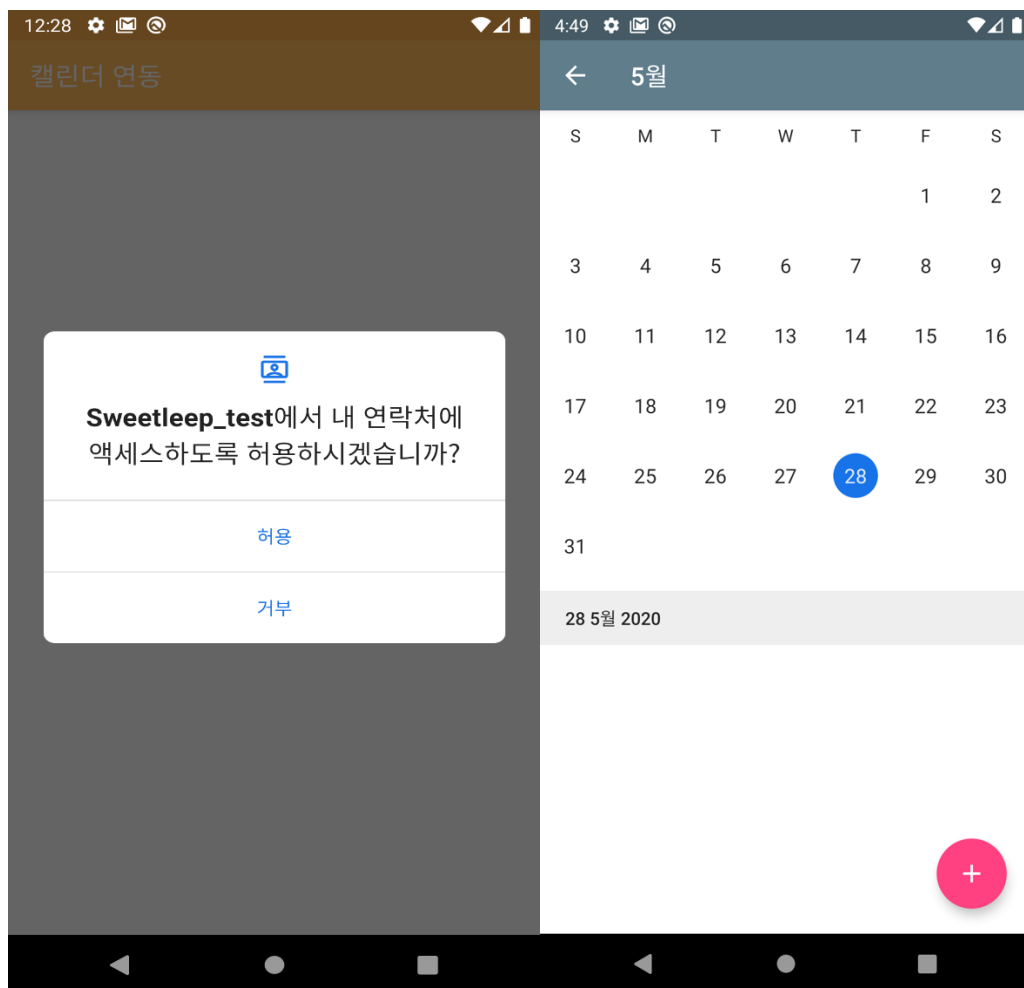


 국민대학교 소프트웨어학부 캡스톤 디자인 I	중간보고서(2차)		
	프로젝트 명	Smarlarm(스말람)	
	팀 명	Sweetleep	
	Confidential Restricted	Version 1.4	2020-05-28

2.2.2 캘린더 기반 알람 생성 서비스

기존 서비스와 차별점을 두기 위한 서비스입니다. 알람 설정을 위해 Google Calendar Api를 이용하여 구글계정과 연동하고, 우리 어플용 캘린더를 생성, 일정 확인 기능을 구현하였습니다.

1) 캘린더 연동



위와 같은 방법으로 접근 권한을 요청하고, 계정이 연동되면 지정된 우리 어플용 캘린더를 생성하고 일정을 불러옵니다. 자신이 원하는 날짜를 선택하여 해당일들의 일정 목록을 불러오며 시작 시간이 있는 일정의 경우 어플 시작시 입력받은 준비시간을 이용해 알람을 손쉽게 설정할 수 있도록 합니다.

 국민대학교 소프트웨어학부 캡스톤 디자인 I	중간보고서(2차)		
	프로젝트 명	Smarlarm(스말람)	
	팀 명	Sweetleap	
	Confidential Restricted	Version 1.4	2020-05-28

 국민대학교 소프트웨어학부 캡스톤 디자인 I	중간보고서(2차)		
	프로젝트 명	Smarlarm(스말람)	
	팀 명	Sweetleap	
	Confidential Restricted	Version 1.4	2020-05-28

2.2.3 프론트엔드

프론트 엔드에서는 사용자들이 직관적으로 알기 쉽도록 UI를 구성하였습니다. 처음 사용자들이 수면패턴 분석을 사용하는데 어려움이 없도록 앱 처음 실행시에 사용 방법 및 주의사항을 알려줍니다. 이 기능은 어플리케이션 첫 실행 시에만 보여주고 이후 설정 탭을 통하여 다시 볼 수 있도록 만들었으며, 그 외 부분은 탭 이름만 봐도 직관적으로 알 수 있도록 구성하였습니다.

1) 실행화면 Splash Activity

어플리케이션을 실행했을 경우의 화면입니다. 앱이 구동되는 시간 동안 띄워주며 우리 어플의 로고와 이름을 나타냅니다.



 국민대학교 소프트웨어학부 캡스톤 디자인 I	중간보고서(2차)		
	프로젝트 명	Smarlarm(스말람)	
	팀 명	Sweetleep	
	Confidential Restricted	Version 1.4	2020-05-28

2) 튜토리얼 화면

Fragment를 이용하여 사용자가 어떻게 수면패턴 및 알람 설정을 하는지에 대한 튜토리얼을 보여준다. 이 기능은 최초 실행 시 한번만 보여주며 그 이후에는 설정 탭을 통해서 다시 볼 수 있도록 하였습니다. 화면을 슬라이드하거나 NEXT 버튼을 누르면 다음 step으로 넘어가며, SKIP도 가능하도록 했습니다. 마지막 step에서는 GOT IT 버튼을 두어 튜토리얼 화면을 종료하도록 하였습니다.



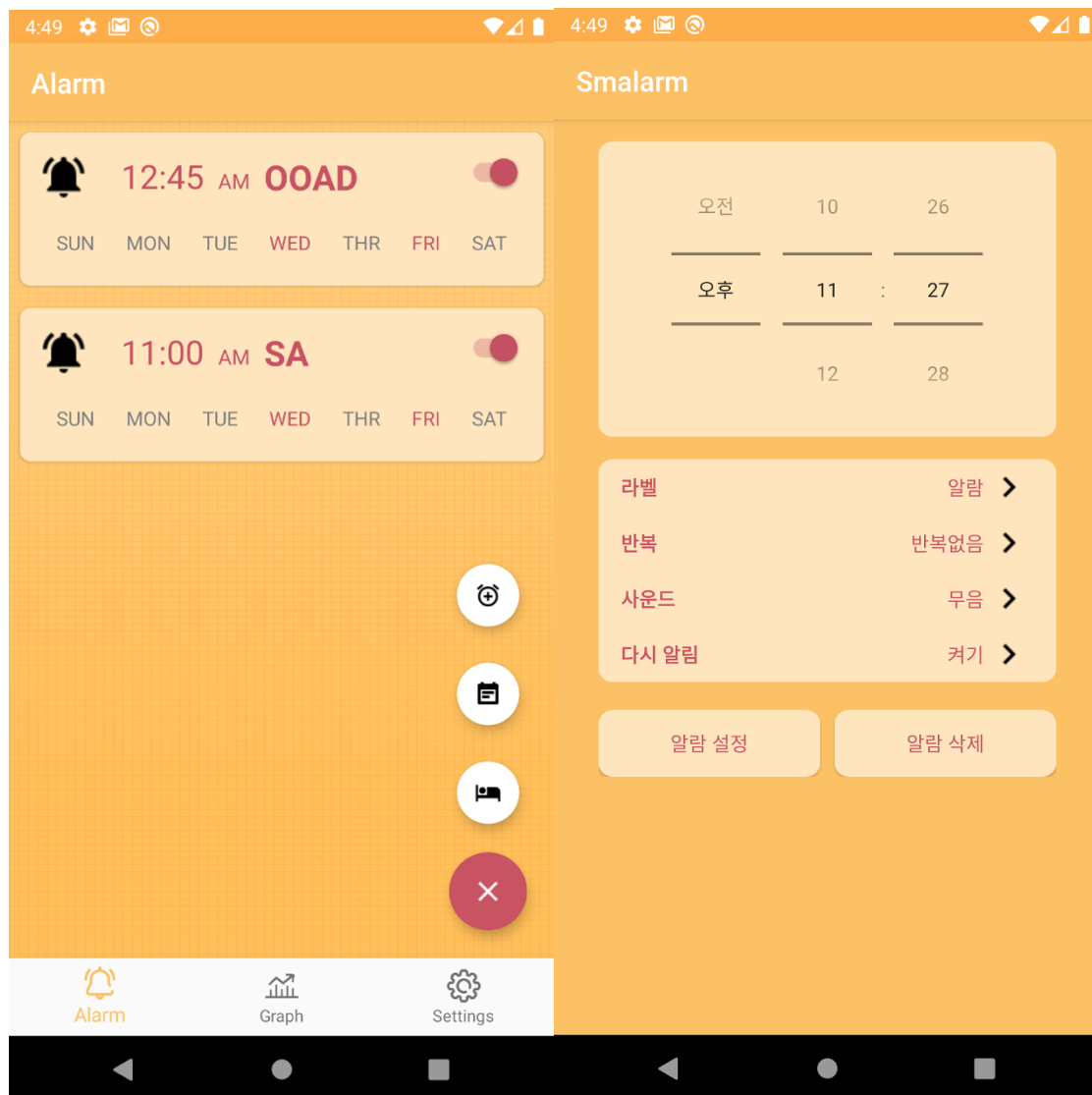
 국민대학교 소프트웨어학부 캡스톤 디자인 I	중간보고서(2차)		
	프로젝트 명	Smaralarm(스말람)	
	팀 명	Sweetleap	
	Confidential Restricted	Version 1.4	2020-05-28

3) 알람 설정 화면 및 알람 리스트 화면

알람 리스트로 홈화면을 대신하였습니다. 알람 설명과 시간, 반복요일을 확인할 수 있습니다. Recyclerview로 구현하여 데이터가 많더라도 view를 재사용하여 빠르게 처리할 수 있도록 했습니다.

오른쪽 아래에 Floating Action Button을 두어 리스트에서 바로 원하는 기능으로 이동할 수 있도록 했습니다. 알람 추가 화면과 캘린더 화면, 수면 분석 시작 기능으로 이동 가능합니다.

알람 설정화면에서는 설정한 시간에 설정한 이름, 사운드로 알람을 울리도록 합니다. 알람을 설정하면 리스트에 자동으로 추가됩니다.



 국민대학교 소프트웨어학부 캡스톤 디자인 I	중간보고서(2차)		
	프로젝트 명	Smarlarm(스말람)	
	팀 명	Sweetleap	
	Confidential Restricted	Version 1.4	2020-05-28

 <div> <p>국민대학교</p> <p>소프트웨어학부</p> <p>캡스톤 디자인 I</p> </div>	중간보고서(2차)		
	프로젝트 명	Smarlarm(스말람)	
	팀 명	Sweetleep	
	Confidential Restricted	Version 1.4	2020-05-28

2.2.4 데이터베이스 & 서버

구글맵 api 호출

```
def call_map():
    client = None
    with open("./key.json","r") as clientJson :
        client = json.load(clientJson)

    #출발지 일단은 고정
    origin      = "37.5728359,126.9746922"
    #도착지
    destination  = "37.5129907,127.1005382"
    #모드 => driving, walking, bicycling, transit
    #모드 => 드라이빙, 도보, 자전거, 대중교통
    mode         = "transit"
    #출발시간
    departure_time = "now"
    #키값
    key          = client["key"]

    url = "https://maps.googleapis.com/maps/api/directions/json?origin="+ origin ₩
        + "&destination=" + destination ₩
        + "&mode=" + mode ₩
        + "&departure_time=" + departure_time₩
        + "&language=ko" ₩
        + "&key=" + key

    request      = urllib.request.Request(url)
    context      = ssl._create_unverified_context()
    response     = urllib.request.urlopen(request, context=context)
    responseText = response.read().decode('utf-8')
    responseJson = json.loads(responseText)

    #구글맵 api 결과 json return
    return responseJson
```

 국민대학교 소프트웨어학부 캡스톤 디자인 I	중간보고서(2차)		
	프로젝트 명	Smarlarm(스말람)	
	팀 명	Sweetleap	
	Confidential Restricted	Version 1.4	2020-05-28

해당 디렉토리에 있는 key.json파일을 사용하여 구글맵 api를 확인하고 구글맵 api에 요청을 보낸다. 임시로 출발지와 도착지 mode(교통수단)은 고정을 시켜놓아서 사용하였다. 얻은 json을 사용하여 사용자가 설정한 이동시간 / 기상시간 / 준비시간을 고려하여 결과값을 리턴해준다.

판단 방법은 크게 4가지를 사용할 예정이다.

```
# 1.현재시간, 출발시간의 차이가 5 분이상이면 무조건 0 을 return
# 2.현재시간, 출발시간의 차이가 5 분이하라면
# 3.이동시간+준비시간 > 현재걸리는시간+준비시간 -> 원래시간으로 일어나도된다
# 4.이동시간+준비시간 < 현재걸리는시간+준비시간 -> 예정보다 일찍 일어나야한다.
```


 국민대학교 소프트웨어학부 캡스톤 디자인 I	중간보고서(2차)		
	프로젝트 명	Smarlarm(스말람)	
	팀 명	Sweetleap	
	Confidential Restricted	Version 1.4	2020-05-28

3. 수정된 연구 내용 및 수정사항

3.1 수정된 내용

3.1.1 기존 시스템과 차별점

기존 시스템과의 차별점이 가장 중요한 요소 중에 하나라고 볼 수 있습니다. 그러한 차별점은 저희는 수면방해요소에서 찾았습니다. 알람 맞추기 직전에 수면에 방해되는 요소를 가볍게 터치하여 사용자의 수면 방해 요소로 그래프를 클래스 분류하여 보여 줍니다. 그렇게 되면 사용자마다 각각 어느 요소가 가장 방해되는지 알게 되고 그에 행동을 줄이게 유도해줍니다.

수면 분석에 추가적으로 스마트 알람 서비스를 제공합니다. 캘린더 일정과 연동한 뒤 일정 시작 시간과 도착지의 정보와 교통수단을 알게 되면 그날의 교통상황을 예상하여 조금 더 일찍 일어나야할지 평소보다 여유를 가져도 될지에 따라 유동적으로 알려줍니다.

또한 기상 알람 뿐 아니라 수면 알람 기능을 구현하여 적정 수면 시간을 기준으로 부족하다 판단될 경우 역으로 잠 잘 시간이라는 알람을 울리도록 합니다.

3.1.2 깊은 숙면 유도

사람의 5가지 감각을 통해서 깊은 숙면을 유도할 수 있다고 합니다. 저희는 핸드폰 어플리케이션이기에 오감 중 청각을 채택하여 깊은 숙면을 유도해줍니다. 통계적으로 사람들이 편안하게 느끼는 소리들을 담아 수면패턴분석시에 노래가 나올 수 있도록 합니다. 잠이 들었다고 판단되면 노래는 자연스럽게 종료됩니다. 이러한 노래에는 빗소리, 바다소리, 산소리 등 여러가지 중에 사용자가 설정탭에서 선택할 수 있도록 합니다. 다른 소리의 경우 사용자 개인이 업로드하여 사용할 수 있도록 합니다.

 국민대학교 소프트웨어학부 캡스톤 디자인 I	중간보고서(2차)		
	프로젝트 명	Smarlarm(스말람)	
	팀 명	Sweetleep	
	Confidential Restricted	Version 1.4	2020-05-28

4. 향후 추진 내용

4.1 GPS연동

스마트폰의 위치서비스를 사용하여 사용자 위치를 파악하여 목적지까지 걸리는 시간을 더욱 정확히 계산한다. 그렇게 되면 출발지가 변경되더라도 서비스 제공이 가능하고, 목적지까지 걸리는 시간과 교통상황에 따른 알람 시간 조정이 가능합니다.



 <div> 국민대학교 소프트웨어학부 캡스톤 디자인 I </div>	중간보고서(2차)		
	프로젝트 명	Smarlarm(스말람)	
	팀 명	Sweetleap	
	Confidential Restricted	Version 1.4	2020-05-28

4.2 날씨 연동

알람을 종료하는 화면에서 사용자가 날씨를 확인하고 우산을 챙겨야 할지 겉옷을 챙겨야 할지에 대한 인지를 할 수 있도록 종료화면에 오늘 날짜, 요일, 시간, 날씨를 표시해준다. 이렇게 되면 잊어버리거나 다시 확인하는 일을 줄여 준비시간 또한 절감할 수 있고 아침시간을 조금 더 효율적으로 사용할 수 있다.



 국민대학교 소프트웨어학부 캡스톤 디자인 I	중간보고서(2차)		
	프로젝트 명	Smarlarm(스말람)	
	팀 명	Sweetleep	
	Confidential Restricted	Version 1.4	2020-05-28

5. 참고문헌

- 심박수, 가속도 센서를 활용한 수면 유도 시스템. 김주연, 김정은, 이승형, 이현수. 한국정보과학회 학술발표 논문집, 2017.06, 1747-1749