



국민대학교
소프트웨어융합대학
소프트웨어학부

캡스톤 디자인 I

종합설계 프로젝트

프로젝트 명	Smarlarm(스말람)
팀 명	<i>Sweetleep</i>
문서 제목	최종보고서

Version	1.1
Date	2020-JUN-08

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

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.


Filename	15조_최종보고서
원안작성자	홍성표
수정작업자	박영민, 박대영, 고민욱, 서민주

수정날짜	대표수정자	Revision	추가/수정 항목	내 용
06-08	박영민	1.0	전체	진행상황에 따른 업데이트
06-09	홍성표	1.1	전체	시스템 구조도 및 목차 변경

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

목 차

1	개요	오류! 책갈피가 정의되어 있지 않습니다.
1.1	프로젝트 개요	오류! 책갈피가 정의되어 있지 않습니다.
1.2	추진 배경 및 필요성	오류! 책갈피가 정의되어 있지 않습니다.
2	개발 내용 및 결과물	오류! 책갈피가 정의되어 있지 않습니다.
2.1	연구/개발 내용 및 결과물	오류! 책갈피가 정의되어 있지 않습니다.
2.2.1	연구/개발 내용	오류! 책갈피가 정의되어 있지 않습니다.
2.2.2	시스템 기능 요구사항	오류! 책갈피가 정의되어 있지 않습니다.
2.2.3	시스템 구조	오류! 책갈피가 정의되어 있지 않습니다.
2.2.4	현실적 제한 요소 및 그 해결 방안	오류! 책갈피가 정의되어 있지 않습니다.
2.2.5	결과물 목록	오류! 책갈피가 정의되어 있지 않습니다.
2.2	기대효과 및 활용방안	오류! 책갈피가 정의되어 있지 않습니다.
3	자기평가	오류! 책갈피가 정의되어 있지 않습니다.
4	참고 문헌	오류! 책갈피가 정의되어 있지 않습니다.
5.	부록	

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

1 프로젝트 추진 목표 및 필요성

1.1.1 프로젝트 개요

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 2 가지 서비스를 중점적으로 제공합니다..

1.1.2 프로젝트 필요성

본 프로젝트는 현대인들이 자신의 수면 패턴을 파악하고 그에 맞게 스스로 대응 할 수 있도록 도와주고자 합니다.

또한 현대인들에게 알람 서비스는 필수가 되었고 그에 맞게 실시간적으로 변화하는 교통상황 및 날씨를 파악하여 가장 최적의 알람 시간을 추천해 주는 데에 목표가 있습니다.

저희는 3 가지 서비스를 중점적으로 제공합니다.

- 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.

- 수면 중 뒤척임 측정 및 통계제공

자신이 수면 중 얼마나 뒤척이는지 내장된 가속도 센서를 활용해 파악할 수 있습니다. 이를 통해 자신이 몇 시에 얼마나 뒤척이는지를 알 수 있고, 자신의 대략적인 수면패턴을 파악할 수 있습니다. 또한 수면시간을 일별, 주별, 월별, 요일별로 평균값을 제공함으로 써 자신의 수면패턴을 대략적으로 파악할 수 있습니다. 이를 통해 자신의 수면습관을 개선할 수 있는 도우미를 제공하고자 합니다.

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

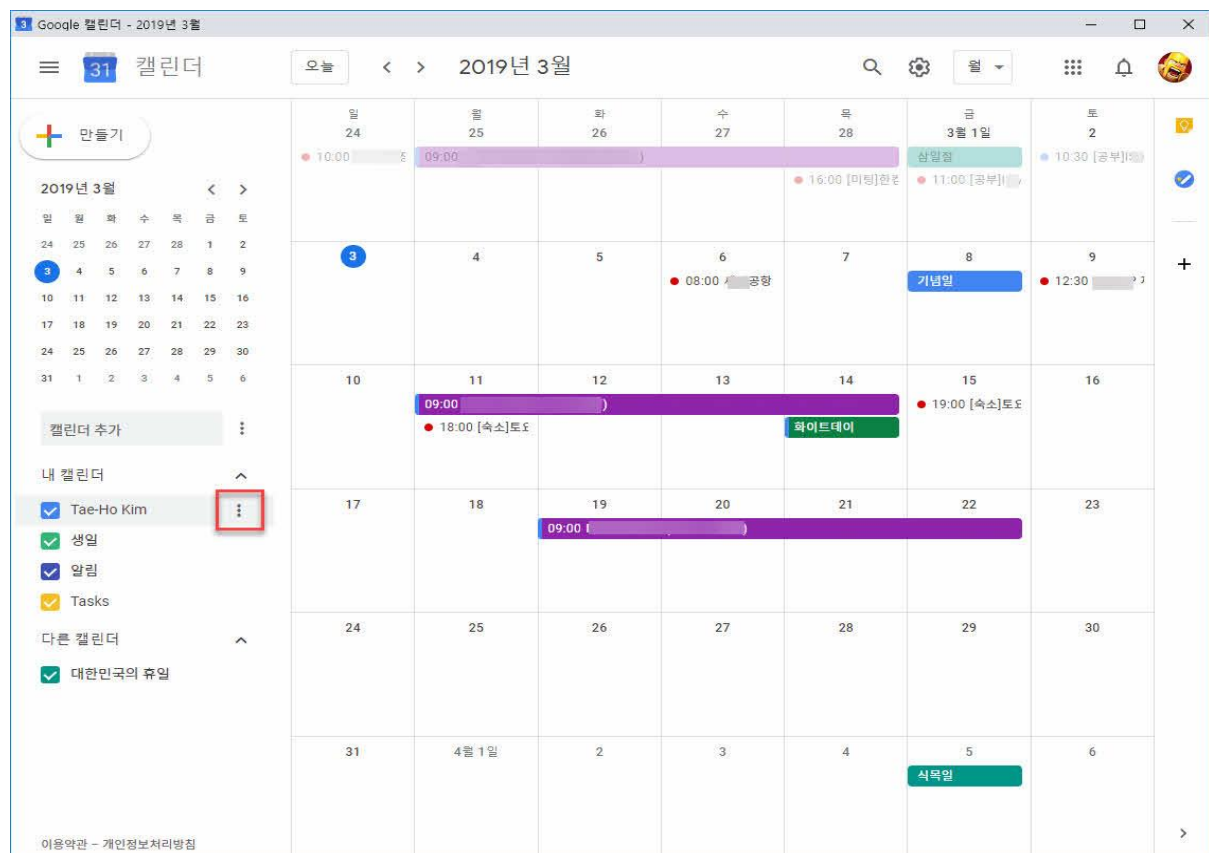
2 개발 내용 및 결과물

2.1.1 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 사용자의 수가 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.



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

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



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.

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

2.1.2 수면 중 움직임 측정 및 수면시간 분석 서비스

스마트폰의 내장된 가속도계 센서를 활용하여, 사용자가 침대위에 스마트폰을 놓고 자면 수면 중 움직임을 측정합니다. 측정된 값은 15분 마다 저장되어서, 그래프로 시각화하여 사용자에게 제공되어 집니다.

또한 수면시작 시간(측정버튼을 누르는시간), 수면종료 시간(알람이 울려서 사용자가 알람을 종료하는 시간)을 기반으로 수면시간을 측정하여, 로컬 DB에 저장 한 후 사용자에게 각 기간별 통계 자료를 보여줍니다. 이를 통해 사용자는 자신의 기간별 대략적인 수면패턴을 파악할 수 있고, 그에 기반해 수면습관을 개선할 수 있을 것입니다. 통계자료를 보여주는 화면은 아래와 같습니다.



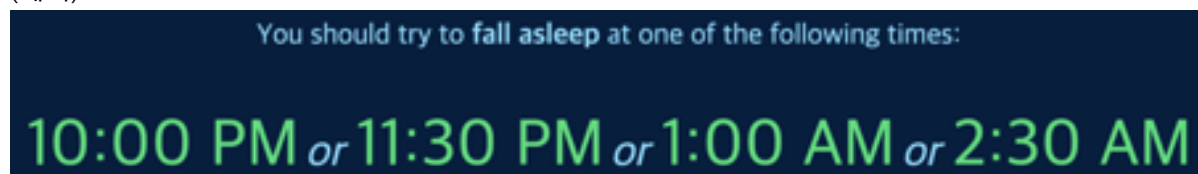
 국민대학교 소프트웨어학부 캡스톤 디자인 I	중간보고서(2차)		
	프로젝트 명	Smarlarm(스말람)	
	팀 명	Sweetleap	
	Confidential Restricted	Version 1.4	2020-05-28

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 캘린더정보 및 수면시간을 저장하는 용도로 사용하기 위함입니다. MySQL를 사용하여 데이터베이스를 만듭니다. 안드로이드에서는 주로 보안상 이유로 직접 접근하지 않고 다른 API 를 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

2.1.6 수면패턴 측정 알고리즘

* 스마트폰 가속도계 측정

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 증가시킵니다. 15분 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;

            // 속도가 설정값보다 크면 움직인 걸로 조건에 걸림 ( 너무 큰 움직임은 측정
            안함 )
            if ((speed > SHAKE_THRESHOLD)) {

                Calendar cal = Calendar.getInstance();
                int minute = cal.get(Calendar.MINUTE);
                int time = 0;
                //long t = cal.getTimeInMillis() / (60 * 1000);
                // int time = (int) t; // 현재시각(분)

                if(0 <= minute && minute < 15){ // 0 ~ 14 분
                    time = (cal.get(Calendar.HOUR_OF_DAY) * 60) + 15;
                }
                else if (15 <= minute && minute < 30){ // 15 ~ 29 분
                    time = (cal.get(Calendar.HOUR_OF_DAY) * 60) + 30;
                }
                else if (30 <= minute && minute < 45){ // 30 분 ~ 44 분
                    time = (cal.get(Calendar.HOUR_OF_DAY) * 60) + 45;
                }
            }
        }
    }
}

```

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

```

else if (45 <= minute && minute < 60){ // 45 분 ~ 59 분
    time = (cal.get(Calendar.HOUR_OF_DAY) * 60) + 60;
}

// 해쉬맵에 저장
this.setMotionCounter(time);
}
lastX = event.values[DATA_X];
lastY = event.values[DATA_Y];
lastZ = event.values[DATA_Z];
}
}
}

```

측정종료 이벤트가 실행되면 onDestroy함수를 실행시켜 다음과 같이 DB에
(Date_id(PK), 수면시간, 측정된 시간, 측정된 움직임횟수) 형식으로 저장합니다.
만약 측정된 값이 2시간 이하이면 DB에 insert하지 않습니다.
여기서 수면시간은
종료시간 (onDestroy() 함수가 call된 시간 = 알람을 off 하는 시간)
- **시작시간** (처음 움직임 값이 저장되었을 시간)으로 계산합니다.

```

@Override
public void onDestroy() {
    // 센서 이벤트 종료
    sensorManager.unregisterListener(SleepSensingService.this);
    // 기록 담기 위한 DB
    SleepDBHelper dbHelper = new SleepDBHelper(getApplicationContext(), "SleepTime.db",
null, 1);

    try {
        // 측정된 15 분단위 time, motionCounter 설정
        setValue();
    }
    catch (Exception e){
        time.add(0);
        motionCounter.add(0);
        e.printStackTrace();
    }

    // 시작, 종료시각 설정
    Calendar cal = Calendar.getInstance();

```



```
startTime = time.get(0);
endTime = (cal.get(Calendar.HOUR_OF_DAY) * 60) + cal.get(Calendar.MINUTE);

int date_id = (int) (cal.getTimeInMillis() / (24 * 60 * 60 * 1000)); // 일
int sleepTime;

String timeJSON = new Gson().toJson(time);
String motionJSON = new Gson().toJson(motionCounter);

try {
    // 측정된 값이 특정조건에 맞을때 ( 2 시간 이상 )
    if ((endTime - startTime) > 120) {

        sleepTime = endTime - startTime;
        //insert data ( sleeptime, time, motioncounter )
        dbHelper.insert(date_id, sleepTime, timeJSON, motionJSON);

    }
}
catch (Exception e){
    e.printStackTrace();
    throw e;
}

super.onDestroy();
}
```

DB 에서 해당날짜 (date_id) 에 해당하는 row를 읽어와서, 사용자가 볼 수 있게 가공하는 메소드는 다음과 같습니다.



```
private void setSleepTimeStat(SleepDBHelper dbHelper, int date_id) {

    // 총 수면시간들의 평균
    int sleep_total = 0;
    int sleep[] = dbHelper.getAllSleepTime();
    for (int i : sleep) {
        sleep_total += i;
    }
    avg_total = sleep_total / sleep.length;

    // 이번주 수면시간 평균
    String week = dateToWeek(date_id);
    int sleep_total_week = 0;
    int[] sleep_week = dbHelper.getWeekSleepTime(week);
    for (int i : sleep_week) {
        sleep_total_week += i;
    }
    avg_week = sleep_total_week / sleep_week.length;

    // 이번달 수면시간 평균
    String month = dateToMonth(date_id);
    int sleep_total_month = 0;
    int[] sleep_month = dbHelper.getMonthSleepTime(month);
    for (int i : sleep_month) {
        sleep_total_month += i;
    }
    avg_month = sleep_total_month / sleep_month.length;

    // 요일별 수면시간 평균
    HashMap<String, Integer> totalSleepTimePerDay = dbHelper.getDayOfWeekSleepTime();//
    //요일별 총 수면시간정보
    HashMap<String, Integer> getDayOfWeekCount = dbHelper.getDayOfWeekCount();// 요일별
    //count 정보

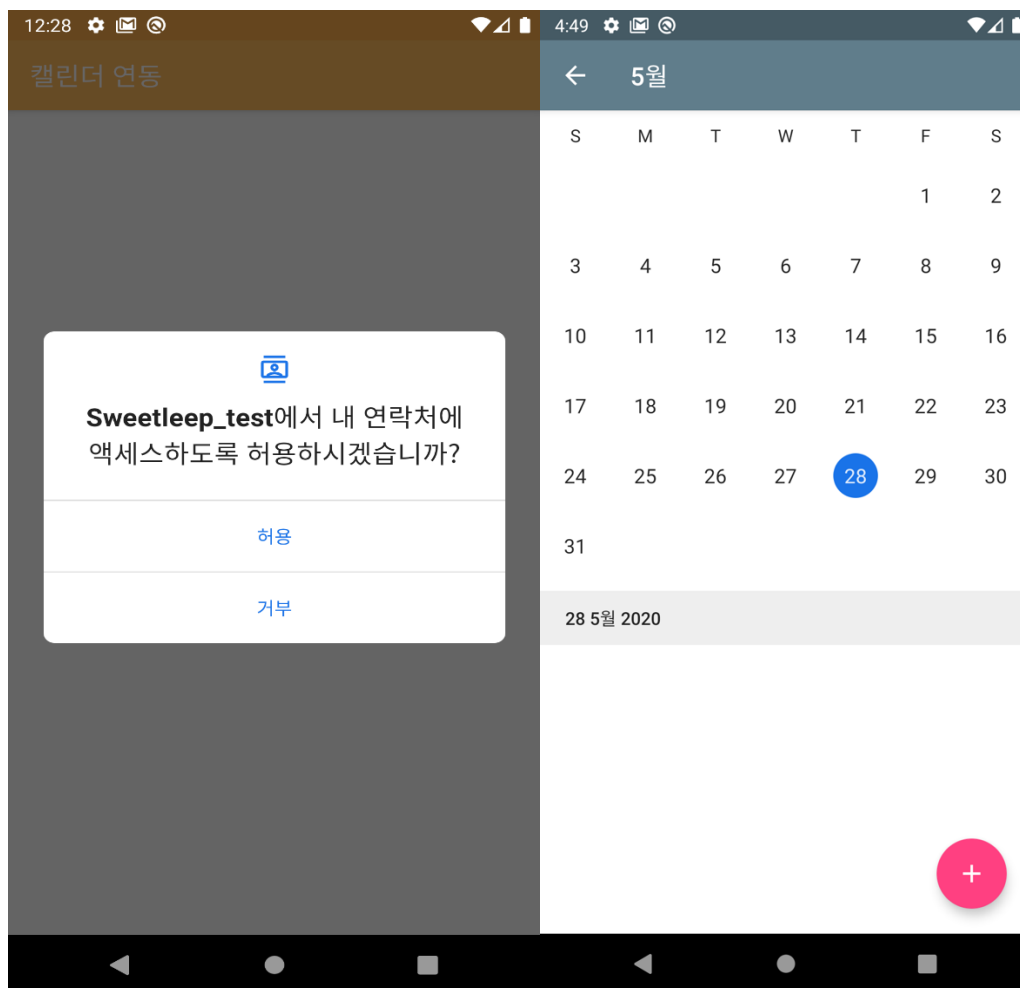
    for (String s : totalSleepTimePerDay.keySet()) {
        int temp;
        temp = totalSleepTimePerDay.get(s) / getDayOfWeekCount.get(s);
        avgSleepTimePerDay.put(s, temp);
    }
}
```

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

2.1.7 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.

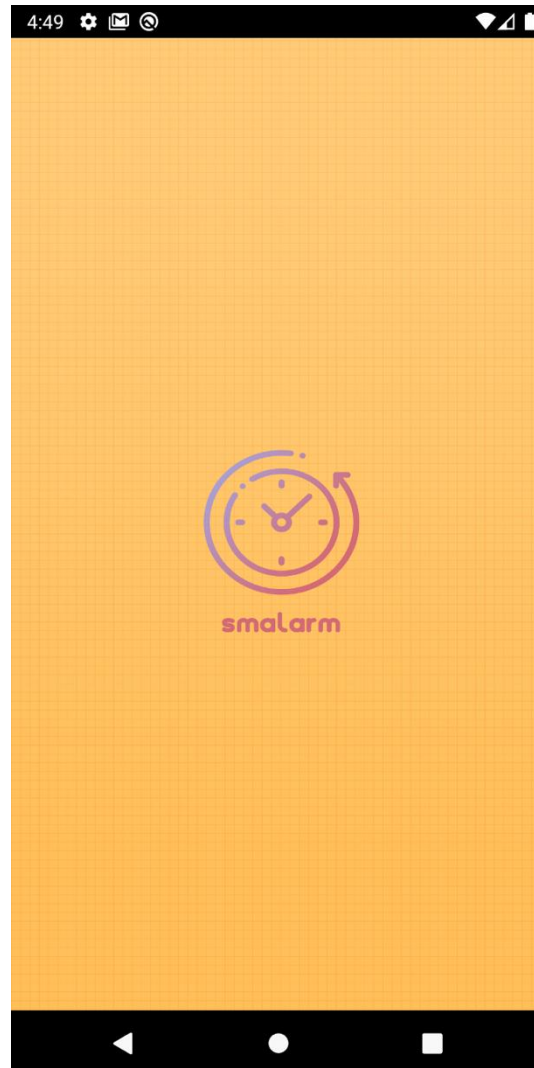
 국민대학교 소프트웨어학부 캡스톤 디자인 I	중간보고서(2차)		
	프로젝트 명	Smarlarm(스말람)	
	팀 명	Sweetleap	
	Confidential Restricted	Version 1.4	2020-05-28


2.1.8 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.



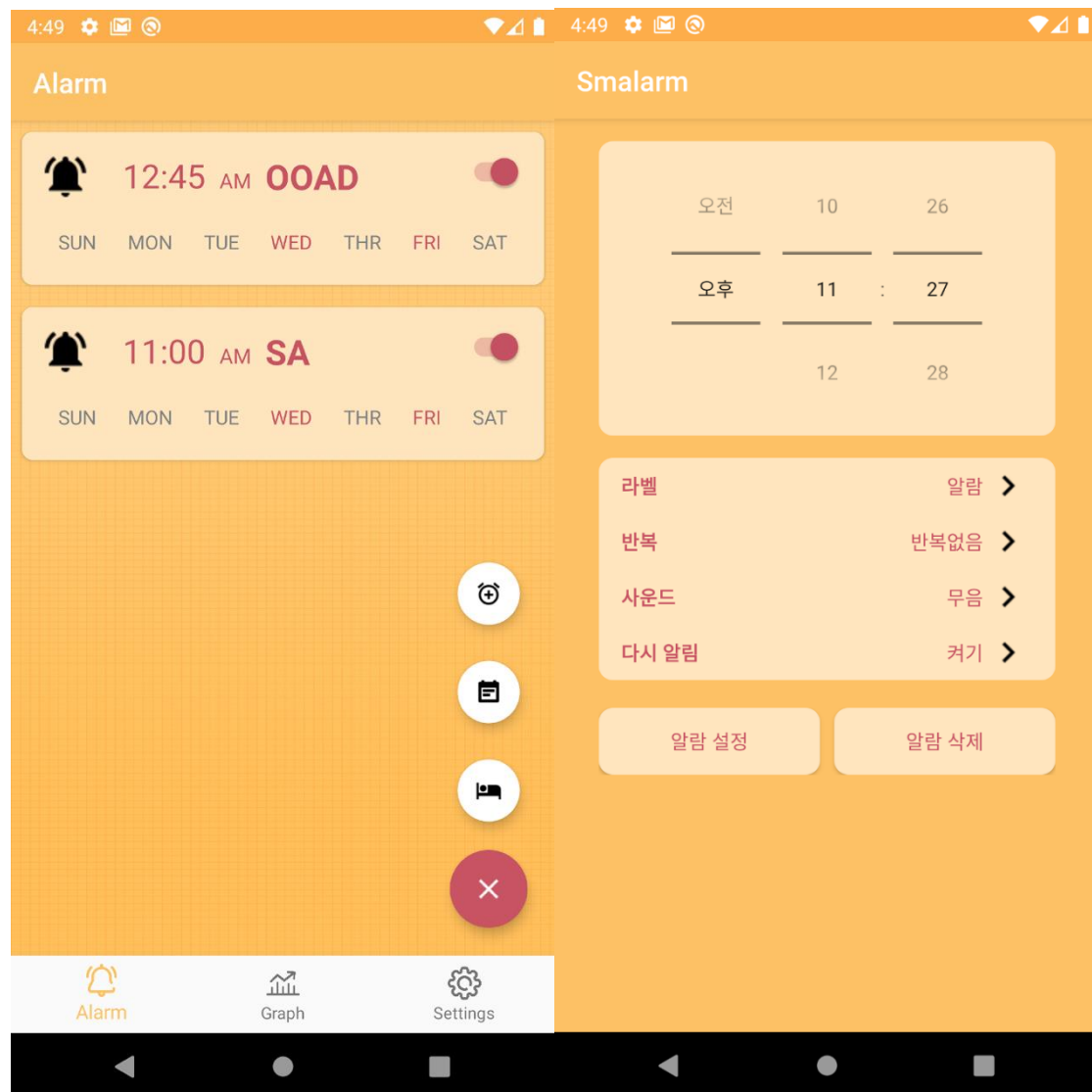
 국민대학교 소프트웨어학부 캡스톤 디자인 I	중간보고서(2차)		
	프로젝트 명	Smaralarm(스말람)	
	팀 명	Sweetleap	
	Confidential Restricted	Version 1.4	2020-05-28

3) 알람 설정 화면 및 알람 리스트 화면

알람 리스트로 홈화면을 대신하였습니다. 알람 설명과 시간, 반복요일을 확인할 수 있습니다. Recyclerview로 구현하여 데이터가 많아도 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.



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

.2.1.9 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

2.2.2 시스템 기능 요구사항

1. Google 캘린더 사용

- 사용자가 구글 캘린더를 이용한다는 점에서 개인의 일정을 관리 하고 있다는 라는 가정하에 일정을 더욱 스마트하게 조절 해줄 수 있게 된다.

2. 잠들기 전 핸드폰 충전

- 사용자의 움직임이나 센서를 사용함에 있어서 배터리 전력이 많이 사용됨으로 사용자는 잠들기 전에 핸드폰 충전기에 핸드폰을 연결한 후 취침에 들도록 한다.

3. 사용자 센서 착용

- 사용자가 스마트 워치가 있으면 더욱 더 정교한 측정이 가능하기 때문에 사용자가 센서를 가지고 있다는 가정하에 수면 패턴 분석을 시작한다. 하지만 사용자가 센서가 없다고 가정하여도 이 기능을 포기 하지 않도록 핸드폰에 있는 자이로 센서로 수면 패턴을 분석한다. 하지만 센서 착용 시 보다는 정확도가 떨어질 수 있다.

4. 스마트 알람

- 사용자가 핸드폰을 통한 알람을 사용한다 라는 점을 요구한다. 핸드폰이 아닌 전자시계나 혹은 아날로그 시계를 사용하여 알람을 맞추지 않고 스마트폰을 이용하여 알람을 맞추는 가정하에 본 어플리케이션을 이용할 수 있다.

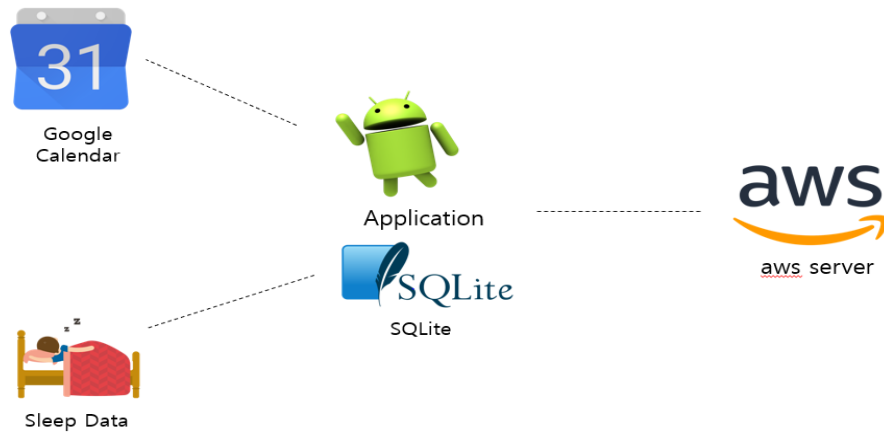
5. 사용자 위치 서비스

- 사용자의 스마트 알람을 위하여 자신의 위치를 알 수 있도록 핸드폰 위치서비스를 사용할 수 있어야한다. 이런점을 바탕으로 출발지에 대한 경도, 위도 정보를 저장한다.

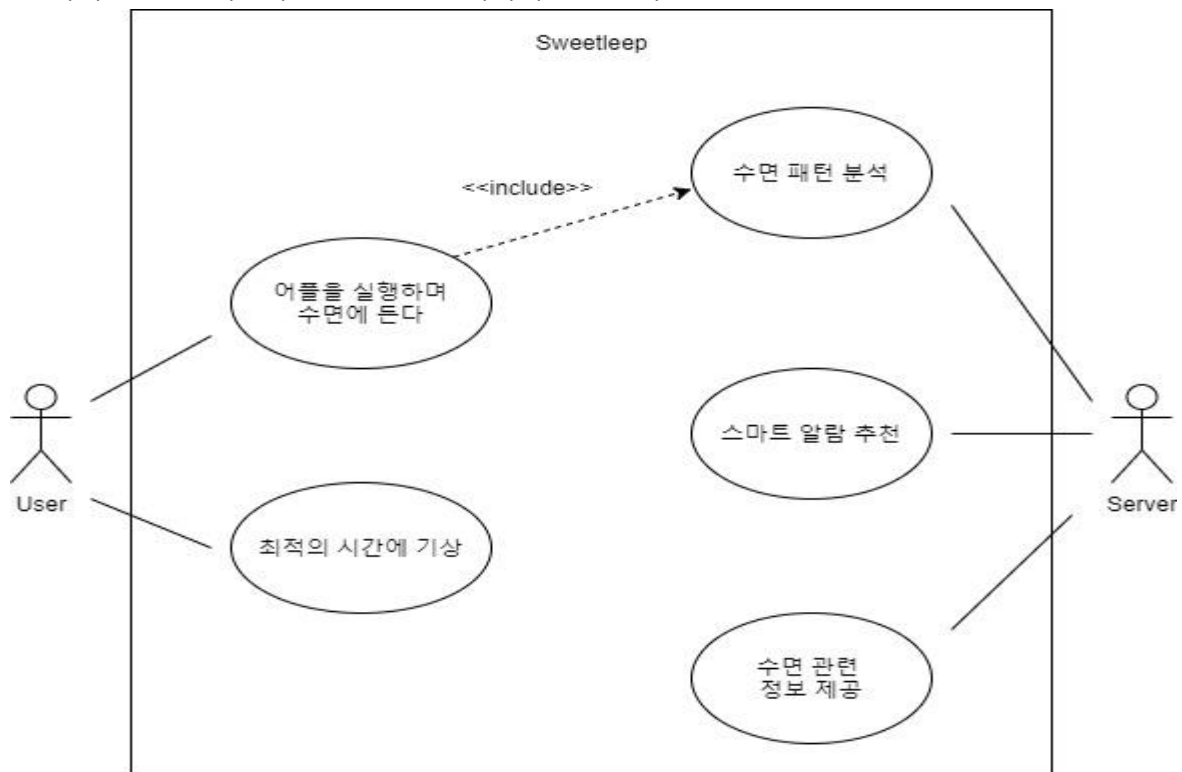
기능적 요구사항	내용	비고
Google 캘린더	구글 캘린더를 통하여 사용자의 일정을 어플리케이션이 알 수 있도록 한다.	완료
스마트 워치	사용자의 움직임을 알 수 있도록 스마트 워치를 착용하면 정확도를 더 올릴 수 있다 하지만 필수 조건은 아니다.	완료
스마트 알람	스마트 알람을 사용할 수 있도록 핸드폰 알람을 이용한다.	완료
위치 서비스	사용자의 스마트폰의 위치를 알 수 있도록 위치서비스를 개방한다.	완료

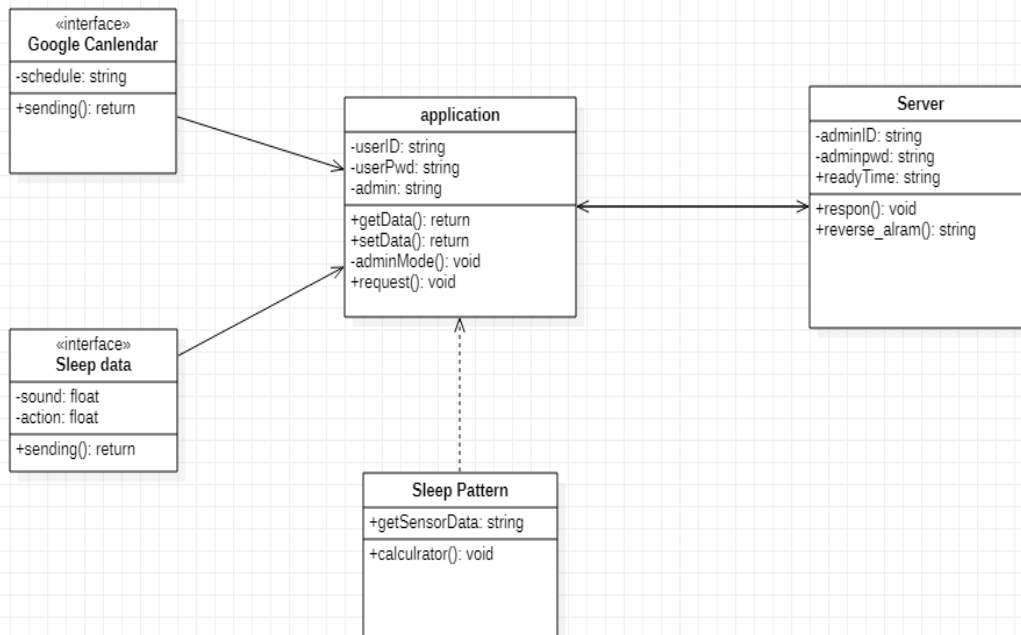
 <div> <p>국민대학교</p> <p>소프트웨어학부</p> <p>캡스톤 디자인 I</p> </div>	중간보고서(2차)		
	프로젝트 명	Smarlarm(스말람)	
	팀 명	Sweetleep	
	Confidential Restricted	Version 1.4	2020-05-28

2.2.3 시스템 구조도

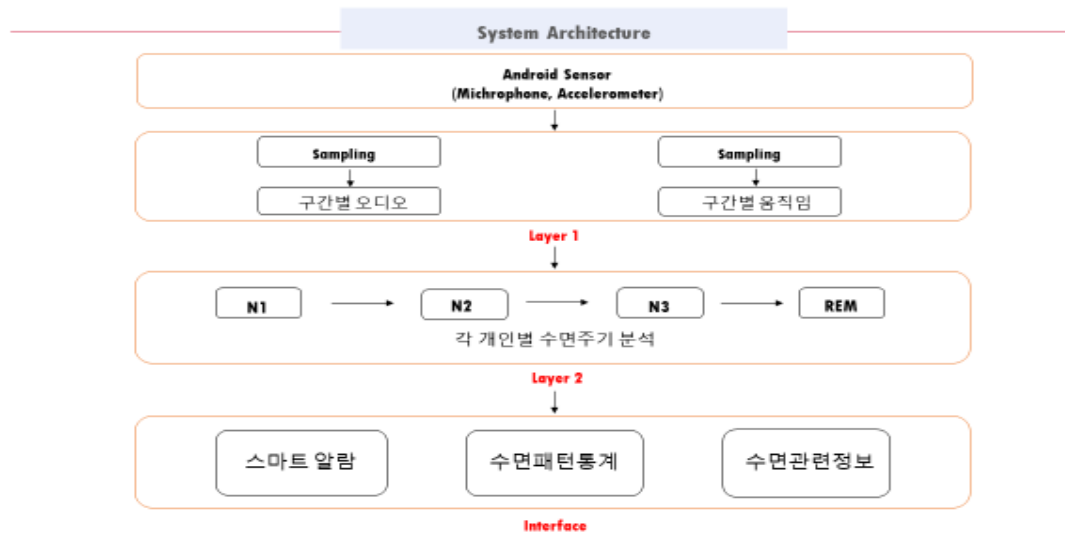



본 프로젝트는 구글 캘린더와 사용자의 수면 데이터를 어플리케이션과 로컬 데이터 베이스를 통하여 정보를 저장하고 그 정보를 서버에 전송한다.





수면 패턴에 대한 시스템 구조도.



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

2.2.4 현실적 제한 요소 및 그 해결방안

1) 수면 패턴 정교화

- 아직 인간의 수면 패턴을 정교화하는 문제가 있다. 소프트웨어 개발자다 보니 인간의 수면에 대한 깊은 이해가 적어 수면 패턴을 정교화하지 못한다. REM 4단계와 깊은 숙면 단계에 대한 이해가 심도 있게 필요하며 이를 수학적으로 계산 할 수도 있어한다. 더불어 수면 패턴을 알기위해서는 뇌파 분석 또한 전문적으로 알아야하며 이 어플리케이션을 정교화 하기 위해서는 수면 전문가가 필요하다.

2) 독립적인 캘린더 사용

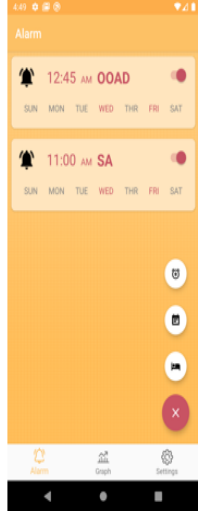
- 현재는 독립적인 캘린더가 아닌 구글 캘린더를 이용하여 사용자의 일정을 받아오고 있다, 하지만 어플리케이션이 정교화 되기 위해서는 독자적인 캘린더를 사용하면 정말 일정과 수면에 대한 필요한 부분을 사용자에게 입력받기가 쉬워짐으로써 더욱 개개인별 사용자에게 맞는 UI와 정보제공을 해줄 수 있다.

3) 독립적인 센서 사용


- 현재는 스마트폰의 센서로 이용하고 있으나, 만약 독립적인 센서를 개발하여 소프트웨어와 연동이 더욱 편리하다면 여러가지 센서를 통하여 수면패턴을 더욱 정확하게 측정할 수 있으며 그렇게 되면 사용자가 자신의 수면 상태를 파악하는데 용이 하게 할 수 있다.

2.2.5 결과물 목록


분류	기능	비고
Application	사용자에게 UI를 제공해주며 수면패턴을 그려주는 기능을 한다.	유
DataBase	수면 패턴을 위한 정보를 저장하고 스마트 알람에 필요한 정보들을 저장한다.	유
Sever	Application에서 정보를 제공받아 교통정보, 날씨등을 계산하여 스마트 알람 기능을 제공하는 기능을 수행한다.	유



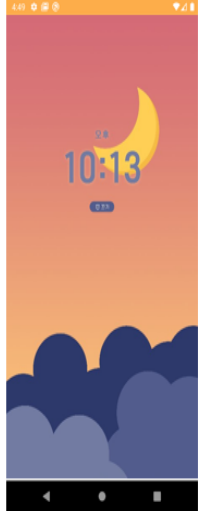
알람 리스트를 보여줍니다.



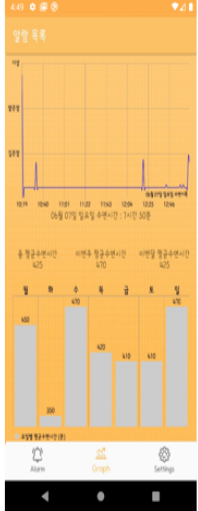
알람 추가 기능을 통해 원하는 시간에 알람을 설정합니다.




Google Calendar를 연동하여 일정을 받아온 후, 일정에 등록된 시작 시간, 반복 요일, 위치 정보로 알람을 설정합니다.



스마트 알람이 켜져있는 경우 교통상황을 고려해 최대 2시간까지 알람을 일찍 올리도록 합니다



또한 스마트폰의 가속도센서를 활용한 수면측정을 통해 수면 중 뒤척임정도 수면시간등을 측정 할 수 있습니다. 측정된 값은 로컬 DB에 저장되며 기간별 수면시간 통계를 제공합니다.

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

2.3 기대효과 및 활용방안

1) 기대효과 및 활용방안

Application을 사용함으로써 효율적인 시간관리와 수면의 질 향상뿐만 아니라 사용자로 하여금 자신의 수면상태를 지속적으로 인지하게 한다.

현대인들은 야근이나 야간자율학습등으로 불면증등과 같은 수면장애들을 겪고 있고 숙면의 중요성 또한 잘 알고있으나 이를 해결하기 위해 실생활에서 행동으로 옮기는 경우는 매우 드물다. 수면의 질을 향상시키기 위한 방법을 모르거나, 안다 해도 이를 위한 스케줄 관리에 어려움을 겪고 바쁜 일상생활중에서 잊어버리는 경우가 많다

이 어플은 지속적인 알림을 주어 사용자가 자신의 수면의 질을 향상시키기 위해 행동해야함을 끊임없이 인지시키고 실제로 실천가능한 방법들을 함께 제공하여 사용자가 직접 행동하게끔 유도하는 효과가 있다.

별도의 하드웨어의 개발없이 기존의 존재하는 슬립테크 기술들과 연계함으로써 더욱 정확한 분석이 가능하고 강화된 숙면효과를 얻을 수 있다. 현재 어플은 센서로부터 사용자의 수면패턴을 분석할 유의미한 값을 받아오는데에 한계가 있는데 현재 여러 기업에서 제공하는 슬립테크 기술들 대부분은 이러한 단점을 해결하기에 아주 좋은 센서를 사용하고 있다. 별도의 센서의 제작없이 이러한 기술들과 어플을 연동하기만 한다면 다양한 상황에서 더 효율적으로 어플을 사용할 수 있을 것이라 예상된다.

3 자기평가

우리는 인생의 1/3을 자면서 보낸다. 밥을 잘 챙겨 먹고, 열심히 운동하는 등 건강관리를 하는 사람들은 많지만, 수면 관리를 하는 사람들은 많이 없다. 하루의 마무리 이자 또 다른 하루의 시작인 잠, 이제는 관리가 필요하다.

잠은 무의식, 무반응, 부동 상태를 의미한다. 그러나 잠꼬대, 뒤척임, 코골이, 이갈이 등의 흔히 잠버릇이라 부르는 증상을 보이는 사람들도 많다. 이는 의학적으로 보았을 때 수면장애의 증상일 수 있다. 무의식 상태에서 나타나는 증상이라 본인이 자각하기는 쉽지 않다.

< 연도별 건강보험 '수면장애' 성별 진료인원 >

(단위: 명, %)

구분	2014년	2015년	2016년	2017년	2018년	'14-'18 증감인원	'14-'18 증감율	연평균 증감율
계	415,502	464,049	495,434	517,654	568,067	152,565	36.7%	8.1%
남성	168,314	190,762	201,828	212,208	239,022	70,708	42.0%	9.2%
여성	247,188	273,287	293,606	305,446	329,045	81,857	33.1%	7.4%
성비	1.5	1.4	1.5	1.4	1.4		-	-

그림1. 연도별 건강보험 '수면장애' 성별 진료 인원

위와 같이 현대인들은 수면장애 및 힘든 일정에 많은 시간과 비용을 많이 사용하고 있다. 이러한 점을 볼때 현대인들의 이러한 점을 해결하기 위한 좋은 시도라고 볼 수 있다. 이를 통하여 모든 문제를 한번에 해결할 수는 없지만 이러한 시도들이 모여서 현대인들의 일정과 수면이 조금 더 나아 질 뿐만 아니라 스마트하게 control 할 수 있지 않을까 기대해보면서 앞으로 더욱 이러한 접근이 많아지는 출발점에서 조금 나아가지 않았을까 생각합니다.

4. 참고문헌

번호	종류	제목	출처	발행년 도	저자	기 타
1	논문	Future Sleep Medicine: Mobile Health and Big Data	Journal of Sleep Medicine	2019	김지영, 조재욱	
2	논문	Sleep Hunter: Towards Fine Grained Sleep Stage Tracking with Smartphones	IEEE	2015	Weixi Gu, Longfei Shangguan, Zheng Yang, Yunhao Liu	
3	논문	Use of mobile phones as intelligent sensors for sound input analysis and sleep state detection.	MDPI	2011	Ondrej Krejcar, Jakub Jirka, Dalibor Janckulik	
4	논문	Analysis of Sleeping Patterns Using Smartphone Sensors	Distributed Computing Group Computer Engineering and Networks Laboratory, ETH Zürich	2011	Steven Meliopoulos, Suhel Sheikh	
5	웹사이트	밤잠 첫 90 분 수면의 질, 다음날 생활의 질 좌우	https://news.joins.com/article/22107604	2017	신윤애 기자	
6	웹사이트	대한수면학회	https://www.sleepnet.or.kr/sleep/normal	—	대한수면학회	

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

5. 부록

서비스 사용법 링크: <https://github.com/kookmin-sw/capstone-2020-15>

영상링크 : <https://www.youtube.com/watch?v=41qOViOwz2k>