**캡스톤 디자인 I**

**종합설계 프로젝트**

|  |  |
| --- | --- |
| 프로젝트 명 | *스마트알람서비스* |
| 팀 명 | *Sweetleep* |
| 문서 제목 | 중간보고서 |

|  |  |
| --- | --- |
| **Version** | 1.4 |
| **Date** | 2020-APIRL-23 |

|  |  |
| --- | --- |
| **팀원** | 홍성표 |
| 고민욱 |
| 박대영 |
| 박영민 |
| 서민주 |
| 양경식 |

|  |
| --- |
| **CONFIDENTIALITY/SECURITY WARNING**  이 문서에 포함되어 있는 정보는 국민대학교 소프트웨어융합대학 소프트웨어학부 개설 교과목 캡스톤 디자인I 수강 학생 중 프로젝트 “**스마트알람서비스**”를 수행하는 팀 “SWEETLEEP”의 팀원들의 자산입니다. 국민대학교 소프트웨어학부 및 팀 “SWEETLEEP”의 팀원들의 서면 허락없이 사용되거나, 재가공 될 수 없습니다. |

**문서 정보 / 수정 내역**

|  |  |
| --- | --- |
| **Filename** | 중간보고서- |
| **원안작성자** | 홍성표 |
| **수정작업자** | 박영민, 박대영 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 수정날짜 | 대표수정자 | Revision | 추가/수정 항목 | 내 용 |
| 2020-04-19 | 홍성표 | 1.0 | 최초작성 |  |
| 2020-04-22 | 박영민 | 1.1 | 수면패턴관련수정 | 작성코드 기록 |
| 2020-04-22 | 박영민 | 1.2 | 프로젝트목표, 수면패턴 | 프로젝트목표(수면패턴) 수정  수면패턴 연구내용 추가 |
| 2020-04-23 | 홍성표 | 1.3 | 내용 수정 및 오타 수정 | 내용 수정 및 오타 수정 |
| 2020-04-23 | 박대영 | 1.4 | 오타 수정 | 오타 수정 |
|  |  |  |  |  |
|  |  |  |  |  |

내용

[1 프로젝트 목표 4](#_Toc38554309)

[2 수행 내용 및 중간결과 5](#_Toc38554310)

[2.1 계획서 상의 연구내용 5](#_Toc38554311)

[2.2.1수면패턴 측정 알고리즘 5](#_Toc38554312)

[2.1.2 알람 추천 서비스 알고리즘 8](#_Toc38554313)

[2.1.3 프론트엔드 10](#_Toc38554314)

[2.2 수행내용 12](#_Toc38554315)

[2.2.1 수면패턴 측정 알고리즘 12](#_Toc38554316)

[2.2.2 알람 추천 서비스 16](#_Toc38554317)

[2.2.3 프론트 엔드 18](#_Toc38554318)

[3. 수정된 연구 내용 및 수정사항 22](#_Toc38554319)

[3.1 수정된 내용 22](#_Toc38554320)

[3.3.1 기존 시스템과 차별점 22](#_Toc38554321)

[3.3.2 깊은 숙면 유도 22](#_Toc38554322)

[4. 향후 추진 내용 23](#_Toc38554323)

[4.1 GPS연동 23](#_Toc38554324)

[4.2 날씨 연동 24](#_Toc38554325)

[5. 참고문헌 25](#_Toc38554326)

# **프로젝트 목표**

본 프로젝트는 현대인들이 자신의 수면 패턴을 파악하고 그에 맞게 스스로 대응 할 수 있도록 도와주고자 한다. 또한 현대인들에게 알람 서비스는 필수가 되었고 그에 맞게 실시간적으로 변화하는 교통상황 및 날씨를 파악하여 가장 최적의 알람 시간을 추천해 주는 데에 목표가 있다. 저희는 3가지 서비스를 중점적으로 제공한다.

#### **- 수면 패턴 분석**

스마트폰에 내장된 가속도계를 이용하여 자신의 수면을 기록한다. 사용자가 수면에 들기 전에 침대위에 스마트폰을 놓고 잠에 든다. 수면 중 뒤척임을 감지하여, 수면 중 자신이 얼마나 움직이는지 그에 따른 수면종류 구분을 시각화해서 나타낸다. 이를 통해 자신의 대략적인 수면주기를 파악할 수 있으며, 이는 개인의 대략적인 수면장애를 파악하는데 도움이 될 것이다.

#### **- 스마트 알람 서비스**

현대인에게 알람은 필수적인 기능이 되었다. 이에 따라 기본적인 알람이 아닌 스마트 알람 기능을 제공한다. Google 캘린더와 연동하여 아침에 일어나 준비하는 시간과 교통수단을 입력 받으면 적당한 알람 시간을 추천해준다. 만약 가는 길이 막힌다면 조금 일찍 그렇지 않다면 조금 늦게 알람 시간을 제공한다.

#### **- 정보제공**

정보제공은 한 달 혹은 일주일 단위로 자신이 과거의 수면 패턴 그래프를 모아서 한 번에 보여주는 기능이다. 이런 기능을 통하여 자신이 어떤 상황에서 잠을 깊이 자는지를 알 수 있으며 혹은 뒤척임, 소리를 통해 사용자의 건강 상태 또한 체크할 수 있다.

더불어 잠을 잘 때 듣기 좋은 노래 혹은 유용한 팁 등을 사용자에게 알려주어 깊은 숙면을 할 수 있도록 유도한다.

# **수행 내용 및 중간결과**

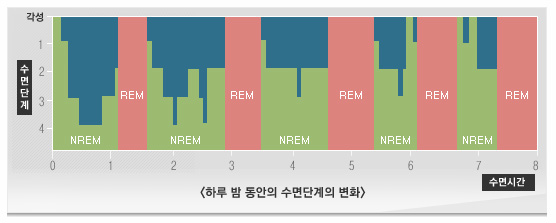
## 계획서 상의 연구내용

### **2.2.1수면패턴 측정 알고리즘**

인간의 수면은 다음과 같은 2가지 수면으로 이루어져 있다.

* **REM수면** : 얕은 잠, 꿈을 꾸는 잠,  동공이 빠르게 움직임
* **N-REM수면** : 깊은 잠, 1 ~ 4단계로 이루어져 있음, 대뇌의 피로가 풀리는 시기

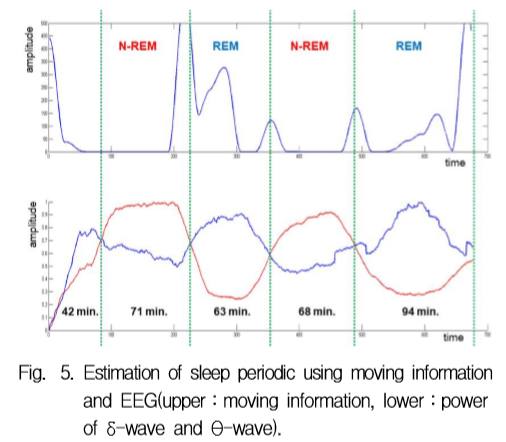
이러한 2가지 수면종류는 다음과 같은 패턴으로 나타나게 된다.



잠든 직후 NREM수면으로 시작해 1단계(얕은 잠)부터 4단계(깊은 잠)까지 진행되며, 각 단계는 5~ 15분입니다. 그 후 REM수면에 진입하게 됩니다. REM수면은 깨기 직전의 얕은 잠이라고 볼 수 있다.

저희는 이러한 인간의 수면패턴에서 REM수면이 끝나고 다시 NREM수면으로 진입할 때, 신체의 상태변화를 주목하고자 한다. 각종 수면관련 연구결과에 따르면 이 시기에는 대뇌의 움직임이 활발한 시기이기 때문에, 그에 수반되는 각종 신체상태가 활발하다고 한다. 그 중 저희는 수면 중 뒤척임을 감지 및 시각화 하여 자신의 대략적인 수면주기를 파악할 수 있는 기능을 구현하려고 한다.

결국 가속도 센서 기반으로 REM수면에서 N-REM수면으로 넘어가는 그 지점을 파악하는 것이 핵심과제가 될 예정이다. 다음 그림은 가속도 기반으로 측정한 REM-NREM 수면 주기 관련 논문 내용 중 일부이다.



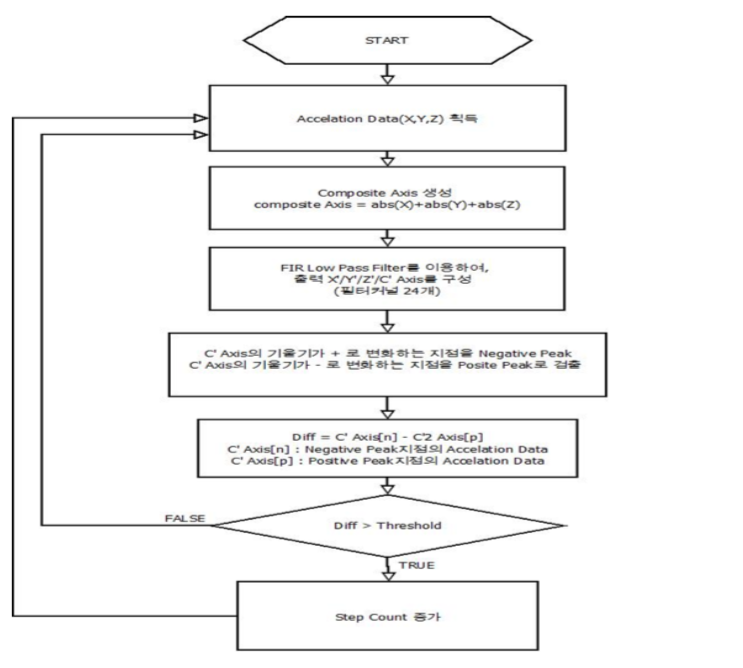
***강경우, 김태선 (2019). 가속도 센서 데이터 기반 수면단계 예측 및 수면주기의 추정. 전기전자학회논문지, 23(4), 158-164.***

그래프 상단은 가속도계 기반으로 한 움직임이며, 하단은 수면분석 DATA 중 하나인 EEG 생체신호의 두가지 뇌파이다. 수면 중 움직임이 적은 N-REM 단계에서는 δ뇌파가 θ뇌파보다 파워가 우세한 것으로 나타나며, 반대로 REM 수면단계에서는 θ뇌파가 δ뇌파 보다 우세한 것으로 나타난다. 같은 구간에서 가속도계 기반 움직임은 REM 수면단계에서는 활발하고, N-REM 수면단계에서는 움직임이 거의 없는 경향을 보인다. 이러한 규칙성을 이용하여 저희는 수면다원검사(PSG) 보다는 정확도는 떨어지지만, 가속도 센서를 통해 수면단계를 구분해보고자 한다.

결국 가속도계 기반으로 측정한 움직임으로 수면종류를 구분하기 위해선, REM-NREM 수면구분을 위한 적절한 임계값(threshold)을 찾는 것이 중요한 과제가 될 것이다.

깊은 잠 시기(NREM수면 4단계)에서 기상을 하게 되면 기상 후에도 잠을 잔 것 같지 않은 그로기(groggy) 상태가 된다. 하지만 대뇌활성도가 높아지는 시기인 (REM수면-NREM수면1단계) 구간에서 기상을 하게 되면 상쾌하게 기상할 수 있다. 이러한 아이디어에 착안하여, 해당 시기에 일어날 수 있도록 도와주는 알람 서비스도 구현할 예정이다.

가속도계를 측정하는 방법은 다음과 같은 알고리즘 모델을 적용하였다.

****

***(김영춘, 조문택, 이충식, 김갑수, 최만규, 주해종. (2015). 수면패턴 모니터링을 위한 가속도 센서처리 알고리즘. 한국산학기술학회 학술대회논문집, (), 373-375.)***

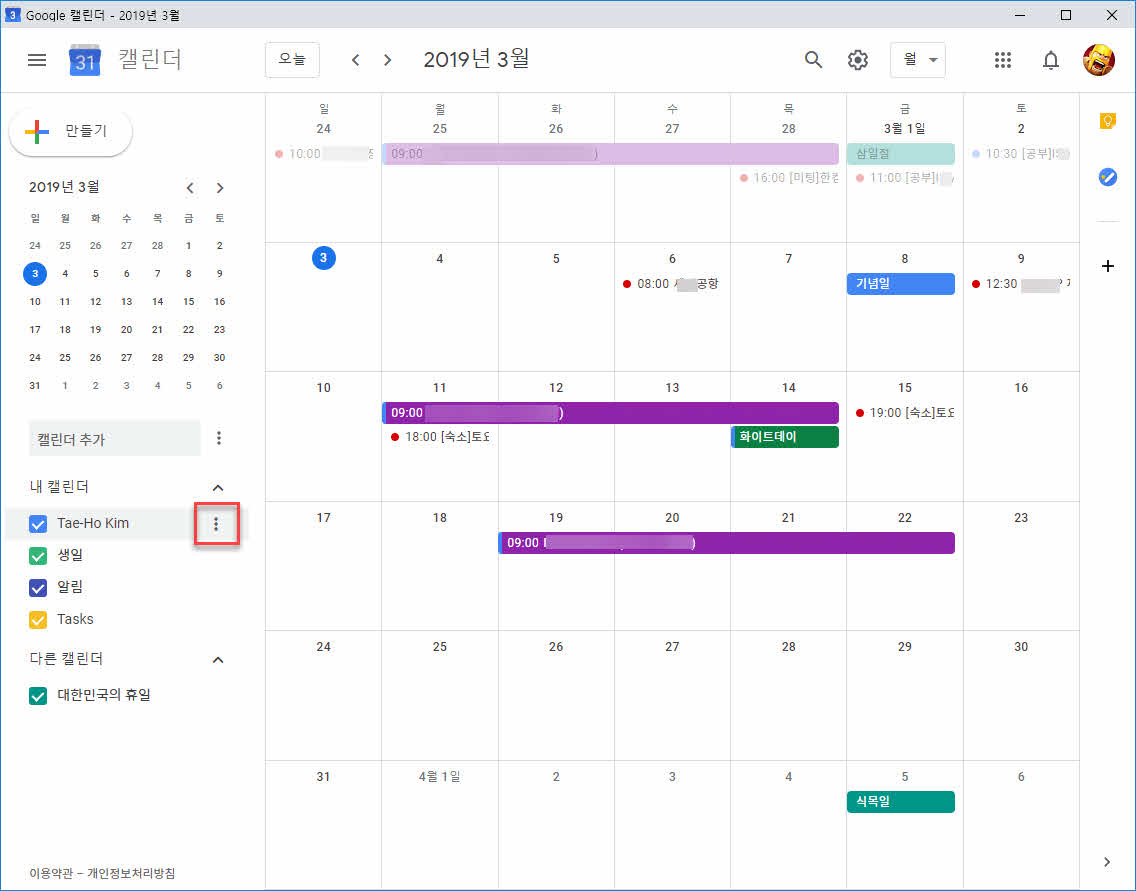
그리하여 저희는 현대인들이 자신의 수면상태를 별도의 장비가 없더라도, 스마트폰만으로도 자

신의 **수면주기를 파악**할 수 있고 그에 기반한 **스마트 알람 서비스**를 구현하고자 한다.

### **2.1.2 알람 추천 서비스 알고리즘**

현대인들에게 시간은 금과 같다. 어떻게 하면 더욱 시간을 효율적으로 사용할 것인지에 대한 고민은 끝없이 진행중에 있다. 이런 점을 조금이나마 덜어주고자 다음과 같이 구현 할 예정이다.

요즘 중, 고등학교 학생들부터 직장인들까지 일정이 많아지고 복잡해짐에 따라 캘린더 어플리케이션을 사용하는 이용자가 많아지고 있다. 이러한 캘린더를 어플리케이션과 연동시켜 어플리케이션 스스로가 사용자의 일정을 알 수 있도록 한다.



위와 같은 캘린더를 연동시킨다. 그 이후 사용자에게 준비시간 및 교통수단을 입력 받으면 사용자가 언제 출발해야 하는지 언제 일어나야 하는지를 최적으로 제공한다.



이 기능의 핵심적 개발은 핸드폰에 있는 캘린더 일정 등록입니다. 연동을 통하여 Google 캘린더와 어플리케이션 간의 정보를 주고 받는다.

public class GoogleCalendar {

private static final String APPLICATION\_NAME = "Google Calendar API Java Quickstart";

private static final JsonFactory JSON\_FACTORY = JacksonFactory.getDefaultInstance();

private static final String CREDENTIALS\_FOLDER = "credentials"; // Directory to store user credentials.

private static final String CALENDAR\_ID = "[캘린더연동아이디]";

자바 코드를 통하여 구글 캘린더를 연동하고 개발을 진행한다. 이렇게 등록된 일정은 어플리케이션을 통하여 푸쉬 알람으로 “알람을 등록 하시겠습니까?” 라는 메시지를 띄우면서 터치를 하면 어플리케이션으로 이동하여 알람을 설정할 수 있도록 도와준다.

### **2.1.3 프론트엔드**

프론트엔드에서는 사용자가 수면패턴분석 사용에 어려움을 느끼지 않도록 어플리케이션 첫 실행 시 사용방법을 알려주는 화면을 구성하여 어려움을 줄이고자 한다.

사용자가 가장 많이 보게 될 화면임으로 최대한 직관적이고 복잡하지 않은 탭들을 구성할 예정이다.

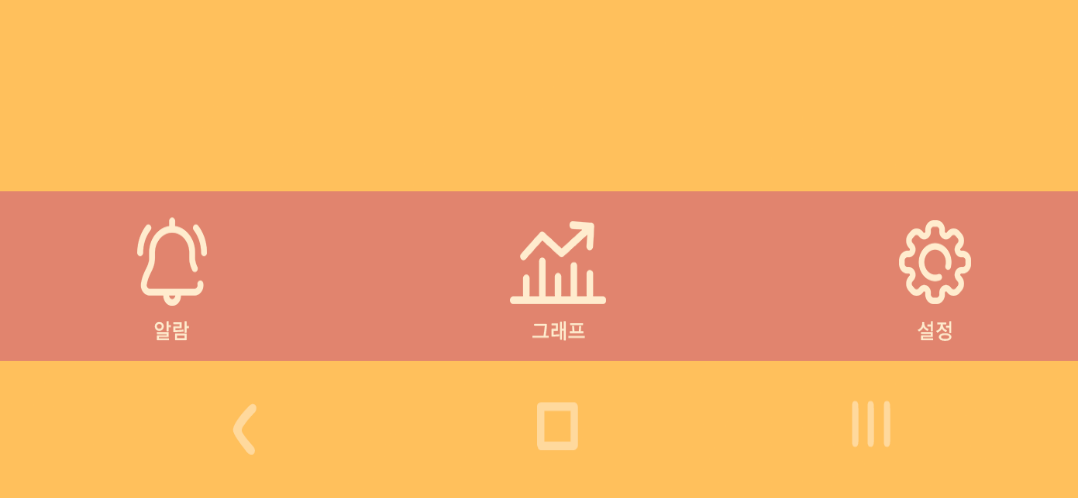
1) 홈화면

홈 화면은 알람이 자주 사용하는 기능이기 때문에 홈 화면을 따로 만들지 않고 아래와 같은 방식의 홈화면 UI를 채탁하여 사용한다.



2) 어플리케이션 탭

탭은 사용자들이 직관적으로 알 수 있도록 다음과 같이 구성한다. 알람 탭을 누르게 되면 알람을 설정할 수 있는 화면으로 이동한다. 그래프를 누를 시에는 지금까지 쌓았던 숙면그래프를 확인할 수 있도록 도와준다. 마지막 설정 탭을 통하여 개인정보 및 알람 소리 등 세부적인 내용을 설정할 수 있도록 도와준다.

****

3) 주의사항

가속도 센싱을 통한 수면패턴그래프 이기 때문에 배터리 소모량이 많아 중간에 핸드폰이 꺼질 수 있는 경우도 생길 수 있다고 생각하여 생각한 UI이다. 수면패턴분석에 들어가면 핸드폰을 꼭 충전기에 연결해주세요 라는 화면을 보여준다.



## 수행내용

### 2.2.1 수면패턴 측정 알고리즘

실시간으로 가속도 좌표를 측정하여, 각 구간별(15분)로 움직임을 카운트하여 데이터로 표현해주는 식으로 구현하였다. 각 구간은 테스트를 통하여 정확도를 높이는 방향으로 설정할 예정이다. 아직 각 수면단계를 구분하는 부분은 정확히 구현하지 못했다.

해당 서비스는 안드로이드 백그라운드서비스에서 측정되야 하므로 다음과 같은 method로 백그라운드 서비스를 실행시킨다. SDK 버전 26이상부터는 백그라운드 서비스가 막혀 있으므로, Notice를 통한 foreground 서비스를 호출한다.

public void startService() {  
  
 Intent serviceIntent = new Intent(this**,** MyService.class)**;** serviceIntent.putExtra("inputExtra"**,** "수면 중 뒤척임 감지중입니다.")**;** if (Build.VERSION.*SDK\_INT* >= **26**) {  
 ContextCompat.*startForegroundService*(this**,** serviceIntent)**;** }  
 else {  
 startService(serviceIntent)**;** }  
}

움직임이 감지된 경우 다음과 같은 onSensorChanged() 메소드에서 측정된 가속도계 좌표를 기반으로 속도(speed)를 구한 후 그 차이를 측정한다. 그 속도가 threshold(민감도) 보다 높다면, 의미 있는 움직임이라 보고 setMotioncounter() 메소드를 통해 해당 구간의 움직임 횟수를 증가시켜준다. 구간의 길이는 15분이며, 데이터 저장 형식은 hashmap을 사용하였다.

@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*) ){  
  
 // 15분 단위로 할때  
 Calendar cal = Calendar.*getInstance*()**;** int minute = cal.get(Calendar.*MINUTE*)**;** int time = **0;** if(**0** <= minute && minute < **15**){ // 0 ~ 14 분  
 time = (cal.get(Calendar.*HOUR\_OF\_DAY*) \* **60**) + **15;** }  
 else if (**15** <= minute && minute < **30**){ // 15 ~ 29분  
 time = (cal.get(Calendar.*HOUR\_OF\_DAY*) \* **60**) + **30;** }  
 else if (**30** <= minute && minute < **45**){ // 30분 ~ 44분  
 time = (cal.get(Calendar.*HOUR\_OF\_DAY*) \* **60**) + **45;** }  
 else if (**45** <= minute && minute < **60**){ // 45분 ~ 59분  
 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*]**;** }  
 }  
}  
// 해당 구간별 움직임을 담는 메소드  
public void setMotionCounter(int time){  
 if(countHash.get(time) == null){  
 countHash.put(time**, 1**)**;** }  
 else {  
 int temp = countHash.get(time)**;** if(temp < **40**) {  
 countHash.put(time**,** temp + **1**)**;** }  
 }  
}\

측정한 구간별 움직임 데이터를 보정작업 (빈값 메꾸기 등)을 거쳐서, 다음과 같은 ArrayList로 변환 후 각 printInfo(), draw() 메소드를 통해 데이터를 표현해준다.

ArrayList<Integer> time = (ArrayList<Integer>) intent.getSerializableExtra("time")**;**ArrayList<Integer> motionCounter = (ArrayList<Integer>) intent.getSerializableExtra("motionCounter")**;**ArrayList<Integer> preTime = (ArrayList<Integer>) intent.getSerializableExtra("preTime")**;**printInfo(preTime**,** time**,** motionCounter)**;** // 수면시간, 수면질, 구간별 움직임  
draw(time**,** motionCounter)**;**// 그래프

각 수면종류를 판별 및 수면질 계산 부분은 부분은 현재 개발단계에 있으며, 현재는 기상상태를 40 으로 설정 후 (움직임이 없는 0은 **깊은 잠** , 30%에 해당하는 0~12까지를 **얕은 잠**, 70%에 해당하는 13~39까지를 **램 수면**) 으로 설정하여 수면 퀄리티를 계산해보았다.

// 수면질 계산 메소드  
private double setSQ(ArrayList<Integer> motionCounter){  
  
 double SQ**;** for(int i=**0;** i<stepSleep.length**;** i++){  
 stepSleep[i] = **0;** }  
  
 for(int i : motionCounter){  
  
 if( i==**0** ){ // 뒤척임 없음 stepSleep[0]  
 stepSleep[**0**] += **15;** }  
 if( (i>**0**) && (i<=**12**) ){ // 뒤척임 적음 stepSleep[1]  
 stepSleep[**1**] += **15;** }  
 if( (i>**12**) && (i<=**39**) ){ // 뒤척임 많음 stepSleep[2] , max 값 40은 기상상태  
 stepSleep[**2**] += **15;** }  
 }  
 SQ = (((stepSleep[**2**]\***0.5**) + (stepSleep[**1**]\***0.75**) + (stepSleep[**0**]))\***100**)/(stepSleep[**0**] + stepSleep[**1**] + stepSleep[**2**])**;** return SQ**;**}

그래프는, 안드로이드 차트 API 중 가장 유명한

MPAndroidChart (<https://github.com/PhilJay/MPAndroidChart>) 를 이용하였다.

LineData data = new LineData(xaxes**,**lineDataSet1)**;**lineChart.setData(data)**;**lineChart.setVisibleXRangeMaximum(**65f**)**;**lineChart.getLegend().setEnabled(false)**;**lineChart.setDescription("뒤척임정도")**;**

측정된 데이터를 표현한 결과는 다음과 같다.

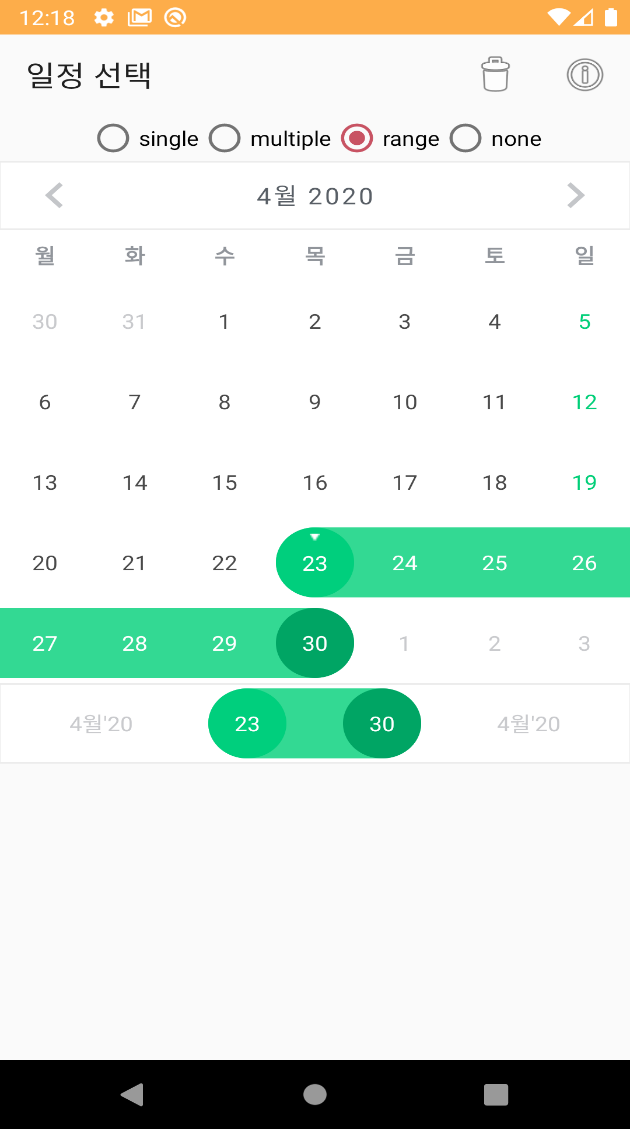
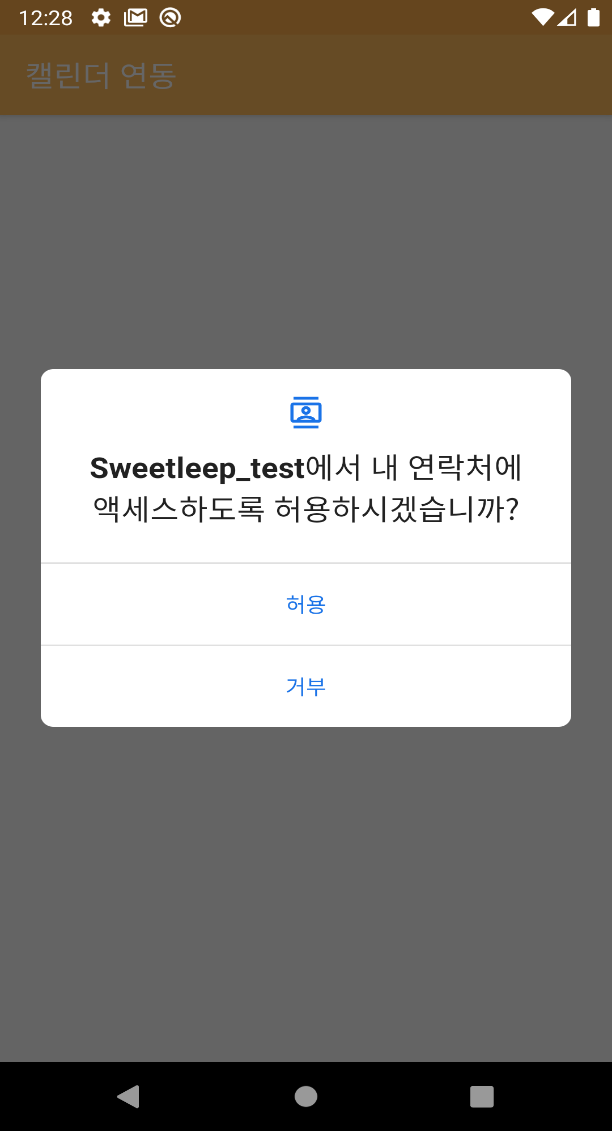


현재는 수면종류분류의 정확성을 보장할 수 없기 때문에, 뒤척임 정도로 표현을 하여 나타냈습니다. 또한 수면 중 구간별 (15분) 움직임도 확인할 수 있다.

### **2.2.2 알람 추천 서비스**

기존 서비스와 차별점을 두기 위한 서비스이다. 알람 시간을 추천해주는 기능을 구현하였다.

1) 캘린더 연동



위와 같은 방법으로 자신이 원하는 일정의 부분을 캘린더와 연동시킨 뒤에 일정의 시작시간을 어플리케이션에 등록한다.

2) 교통수단 및 준비시간

사용자가 준비시간과 교통수단을 입력하게 되면 조건이 충족될 때 일어날 수 있게 조정을 해준다. 만약 미팅이 잡혀 있어 빨리 일어나야 한다면 교통상황을 체크해서 교통체증이 있는 경우 예상보다 조금 더 일찍 일어날 수 있는 알람 시간을 추천해준다.



### **2.2.3 프론트 엔드**

프론트 엔드에서는 사용자들이 직관적으로 알기 쉽도록 UI를 구성하였다.

처음 사용자들이 수면패턴 분석을 사용하는데 어려움이 없도록 앱 처음 실행에 사용방법을 알려주며 이 기능은 어플리케이션 첫 실행 장면에서만 보여준다. 이 후 설정 탭을 통하여 다시 볼 수 있도록 만들었으며 그 외 부분은 탭 이름만 봐도 직관적으로 알 수 있도록 구성하였다.

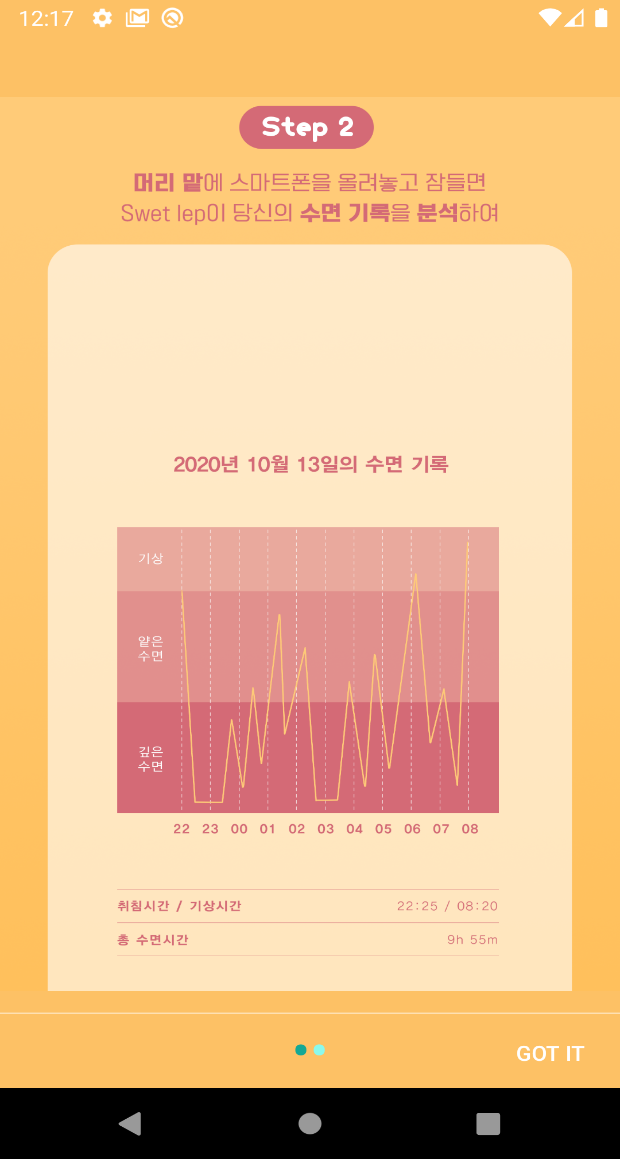
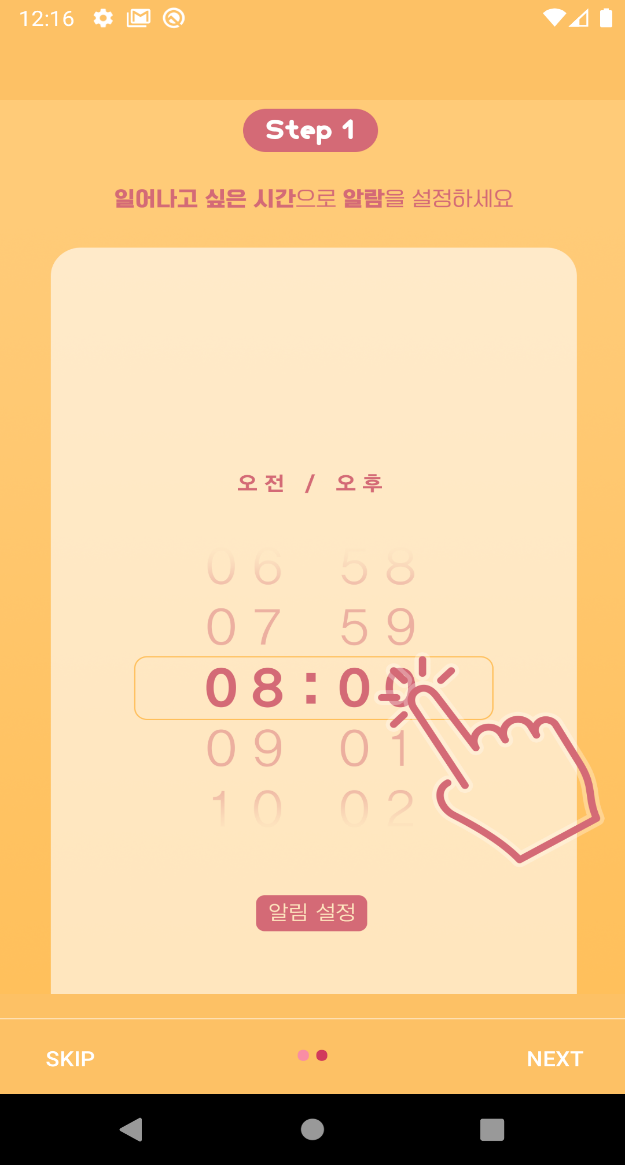
1) 실행화면

어플리케이션을 실행했을 경우의 화면이다.



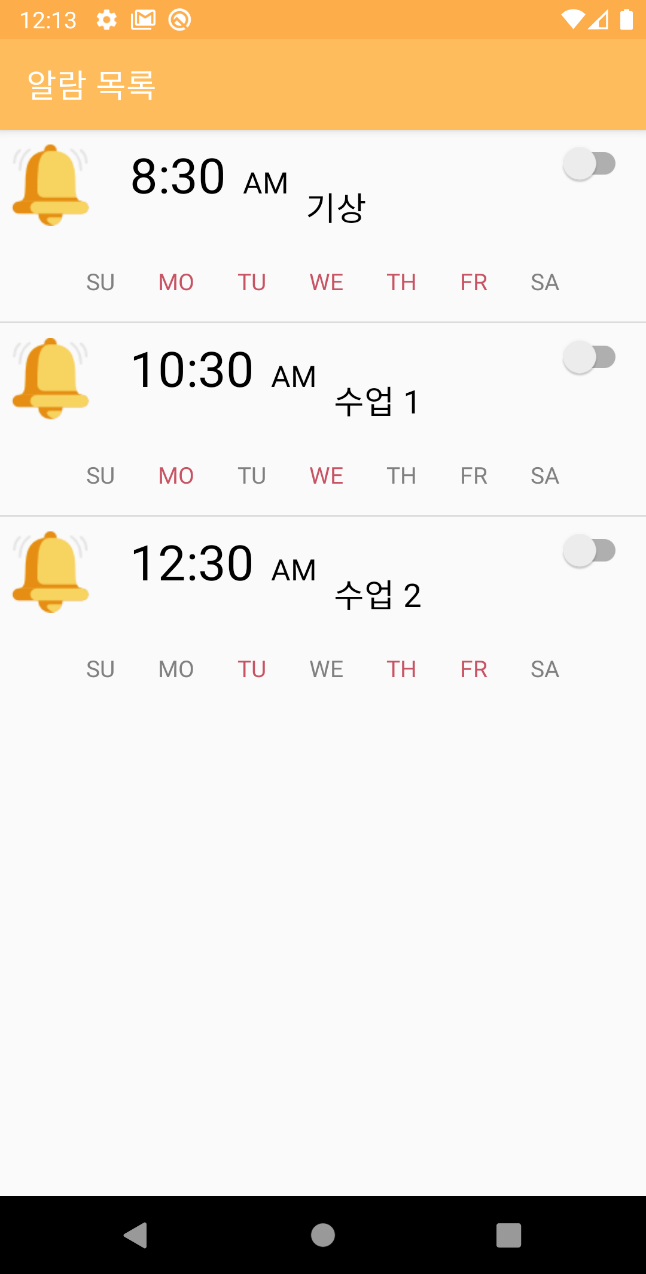
2) 튜토리얼 화면

Step1과 Step2를 통하여 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 하였다.



3) 알람 설정화면

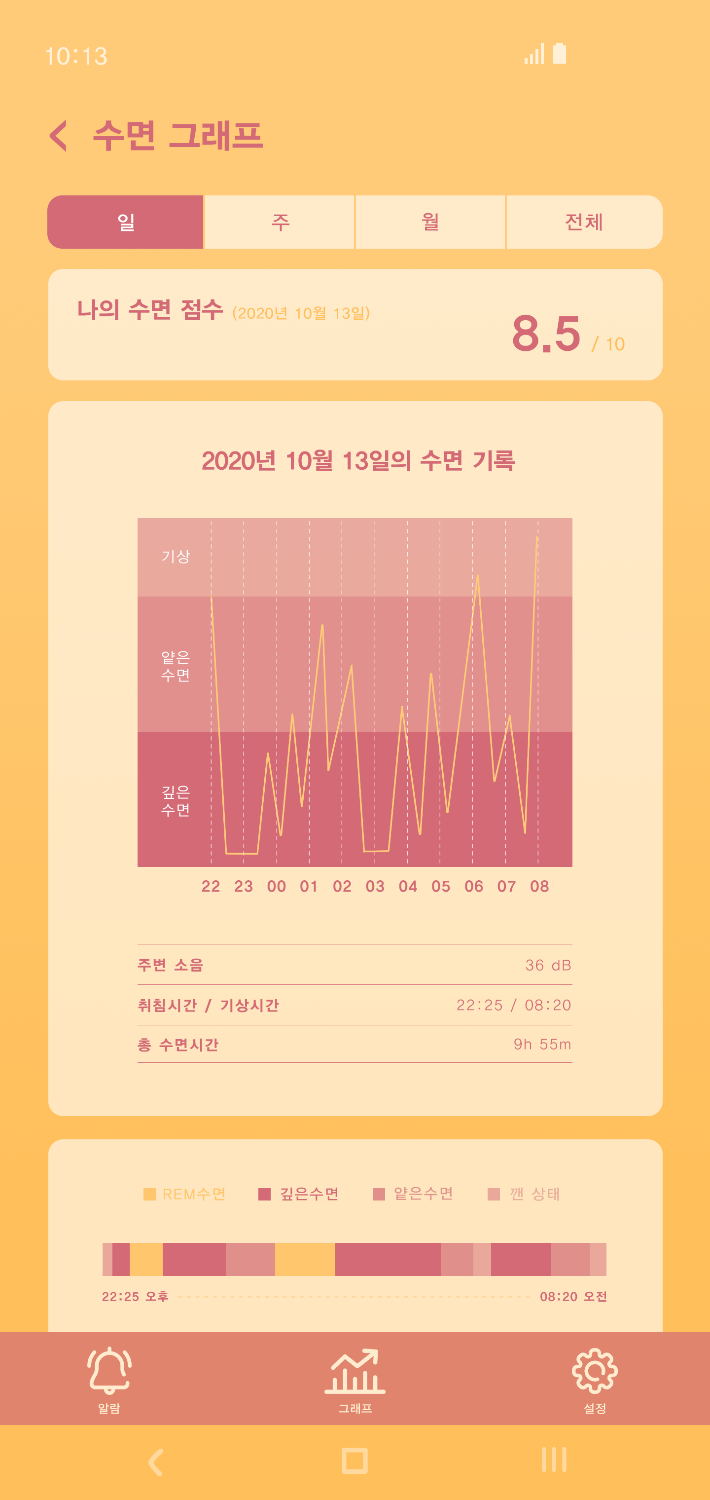
알람 설정화면이다. 내가 원하는 시간을 알람을 설정한다. 설정한 알람은 알람 목록에 저장되어 알람을 on/off 할 수 있다,

****

4) 그래프

그래프 탭을 이용하여 나의 수면 점수 및 수면 그래프를 확인할 수 있는 화면이다.

주변 소음 및 취침시간/기상시간 등을 쉽게 확인할 수 있으며, 일별 주별 월별을 선택하여 확인할 수도 있다.



3. 수정된 연구 내용 및 수정사항

## 3.1 수정된 내용

## 3.3.1 기존 시스템과 차별점

기존 시스템과의 차별점이 가장 중요한 요소 중에 하나라고 볼 수 있다. 그러한 차별점은 우리는 수면방해요소에서 찾았다. 알람 맞추기 직전에 수면에 방해되는 요소를 가볍게 터치하여 사용자의 수면 방해 요소로 그래프를 클래스 분류하여 보여준다. 그렇게 되면 사용자마다 각각 어느 요소가 가장 방해되는지 알게 되고 그에 행동을 줄이게 유도해다.

수면패턴분석에 추가적으로 알람 서비스를 제공한다. 캘린더 일정과 연동한 뒤 일정시작 시간과 도착지의 정보와 교통수단을 알게 되면 그날의 교통상황을 예상하여 조금 더 일찍 일어나야 할지 평소보다 여유를 가져도 될지에 따라 유동적으로 알려준다.

3.3.2 깊은 숙면 유도

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

4. 향후 추진 내용

## 4.1 GPS연동

사용자 어플리케이션에 위치서비스를 사용하여 위치를 정확하게 파악하여 걸리는 시간을 더욱 상세히 계산한다. 그렇게 되면 출발지를 유동적으로 변경이 가능하다.



4.2 날씨 연동

알람을 종료하고 사용자가 우산을 챙겨야 할지 겉옷을 챙겨야 할지에 대한 인지를 할 수 있도록 종료화면에 오늘 날짜, 요일, 시간, 날씨를 표시해준다. 이렇게 되면 준비시간 또한 절감할 수 있고 아침에 시간을 조금 더 효율적으로 사용할 수 있다.



# **5. 참고문헌**

* **가속도 센서 데이터 기반 수면단계 예측 및 수면주기의 추정.** 강경우, 김태선 (2019). 전기전자학회논문지, 23(4), 158-164.
* **수면패턴 모니터링을 위한 가속도 센서처리 알고리즘.** 김영춘, 조문택, 이충식, 김갑수, 최만규, 주해종. (2015). 한국산학기술학회 학술대회논문집, (), 373-375.