캡스톤 프로젝트 제안서

EMB00000aec676b

|  |
| --- |
| 수면 데이터 기계 학습을 통한  재알람 애플리케이션  팀 명: 들깨쓰 |
| 20176058 박시현  20176758 박진영  20172634 이민희 |

1. **프로젝트 개요**

현대사회에서 사람들이 가장 많이 사용하는 스마트폰의 서비스 중 하나는 알람 기능이다. SAMSUNG Newsroom에서 진행한 <가장 자주 사용하는 스마트폰 기능>에 대한 조사에 따르면, 알람/시계 기능이 3위(3,959명 중 527명)를 차지할 정도로 이용률이 높다. 대부분의 사람들은 잠을 깨기 위해 알람을 순차적으로 맞추어 놓기도 하고 시끄러운 음악을 설정하기도 한다. 아침잠이 많고 알람을 잘 듣지 못하는 사람들을 위해 실제로 다양한 알람 어플이 존재하기도 한다. 하지만 기존의 어플들은 단순히 알람을 끄기 어렵게 하는데 그쳐 있다. 우리는 이러한 점을 보완하여 자동으로 수면을 인식해서 사람들이 추가적인 수고 없이 기상을 완료할 수 있도록 하는 것을 목표로 한다.



그림1. SAMSUGN Newsroom, <https://news.samsung.com/kr/3533>

1. **프로젝트 동기**

현재의 알람 어플을 보면 알람을 끄는 방식을 복잡하고 어렵게 하여 사용자들이 알람을 쉽게 끄고 잘 수 없도록 방지하는 방식을 사용하고 있다. 알람이 울리면 간단한 문제를 풀어야 알람을 끌 수 있는 것을 예로 들 수 있다. 하지만 이렇게 단순히 사용자가 일어나는 데에만 초점을 맞추었을 경우, 그 후에 사용자가 제대로 기상을 한 것이 맞는지 불확실하고 알람을 끄는 이러한 과정이 번거로워서 어플을 사용하지 않는 경우도 적지 않다. 따라서 우리는 사용자가 알람을 끄더라도 그 후에 자동으로 사용자의 수면패턴, 움직임 등을 고려하여 필요시 추가적인 알람을 생성하는 어플을 기획하게 되었다. 또한, 핸드폰 이외의 주변 장치가 없어서 사용자의 별도의 노력이 필요하지 않다는 점이 이 프로젝트의 특징이다.

1. **프로젝트 목표**

**3.1 페르소나 예시**

- 평소보다 일찍 학교에 가야하는 중고등학생 A씨

공부를 하느라 밤 늦게 자고, 아침 일찍 일어나는 A씨는 최근 아침에 일어나는 것에 힘듦을 느끼고 있다. 피곤해 알람을 듣고도 다시 잠에 드는 편이라 평소 부모님이 아침에 깨워 주신다. 하지만 동아리 일, 주번활동, 준비물 구매 등 가끔은 평소보다 빨리 학교에 가야하는 일이 있는 날엔, 이러한 사실을 부모님께 말씀드려 좀 더 일찍 깨워 달라고 부탁하지만, 할 일이 많으셔서 정신이 없으시다 보니 종종 까먹고 A씨를 늦게 깨워 주신다. A씨는 부모님께 화를 내보지만 돌아오는 것은 이젠 본인이 직접 일어나라는 잔소리가 돌아온다.

- 주말 데이트에 항상 늦게 일어나 혼나는 대학생 B씨

계속 일을 미루는 편인 대학생 B씨는 평소 알람을 여러 개를 맞춰 놓고 일어나는 것도 미룬다. 알람을 듣고 일어나서도 침대에 있다 보니 종종 다시 잠에 드는데, 반복 설정을 해 놓은 알람때문에 일어나는 경우도 있지만, 반복 설정에 불편함을 느껴 끈 후, 한 참 뒤 불길한 기분이 들어서 일어나면 지각 확정인 날이 많았다.

- 회사에서 전 날 회식을 한 회사원 C씨

술만 먹으면 아침에 일어나는 게 고역인 C씨는 일어나기 위해 집 안 사진 찍기, 수학 문제 풀기 등 여러가지 어플리케이션을 써봤지만 다 적응을 하고 다시 잠에 들어 버리곤 한다. 다시 잠이 들어도 내가 진짜 일어나야 할 시간에 일어나게 하는 어플리케이션을 찾는다.

- 맞벌이 부부를 도와주는 할머니 할아버지 D씨

할 일을 마치고 낮에 잠깐 깜빡 잠에 들어 유치원 아이 마중을 못 나가곤 해 항상 걱정이다. 주변 사람들이 알람을 맞춰주는 법을 알려주지만, 알람이 오히려 부편하고 번거롭다고 느낀다. 간단하고 확실한 알람 어플리케이션을 찾는다.

이렇듯 남녀노소 제한없이 다양한 사람들이 사용할 것으로 예상된다

**3.2 가정**

이 프로젝트는 크게 두개의 기능 사용자의 상태 분석, 다시 잠에 든 경우 알람에 초점을 두었다. 따라서 사용자가 알람을 듣지 못하고 계속 자는 상황, 기능을 강제 종료하고 더 잠에 듦, 잠에서 뒤척여 핸드폰을 침대밖으로 떨어뜨림 등과 같은 사용 방법에 어긋나는 예외상황이 없다고 가정하였다.

**3.2 최종 목표**

이 프로젝트의 최종 목표는 사용자의 수면 상태 분석을 통한 사용자의 완벽한 기상이다. 수면 상태 분석을 위한 데이터 수집은 추가적인 장치 없이 스마트폰의 어플과 센서만으로 이루어지며, 기계 학습을 통해 사용자의 수면 상태를 인지한다. 이렇듯 인지한 수면상태를 기반으로 사용자가 다시 수면 상태에 빠졌다고 판단이 되었을 경우 알람을 다시 울려 사용자의 기상을 돕는 것이 프로젝트의 목표이다.

1. **개발 배경과 구현 내용**

**4.1 개발 배경**

**- 일상 생활과 가장 밀접한 기기인 스마트폰**

과학기술정보통신부가 발표한 “2018인터넷이용실태조사”에 따르면 모든 가구의 주 평균 인터넷 이용시간이 16시간 30분으로 많은 시간을 인터넷으로 보내는 것을 보였고, 그 중 인터넷 접속 기기로 스마트폰이 94.3%를 차지하는 것을 보였다. 이는 스마트폰이 데스크탑, 노트북, 여러 IoT 기기와 웨어러블 기기 등 다른 기기보다 우리 일상 생활에 훨씬 밀접함을 보였다.

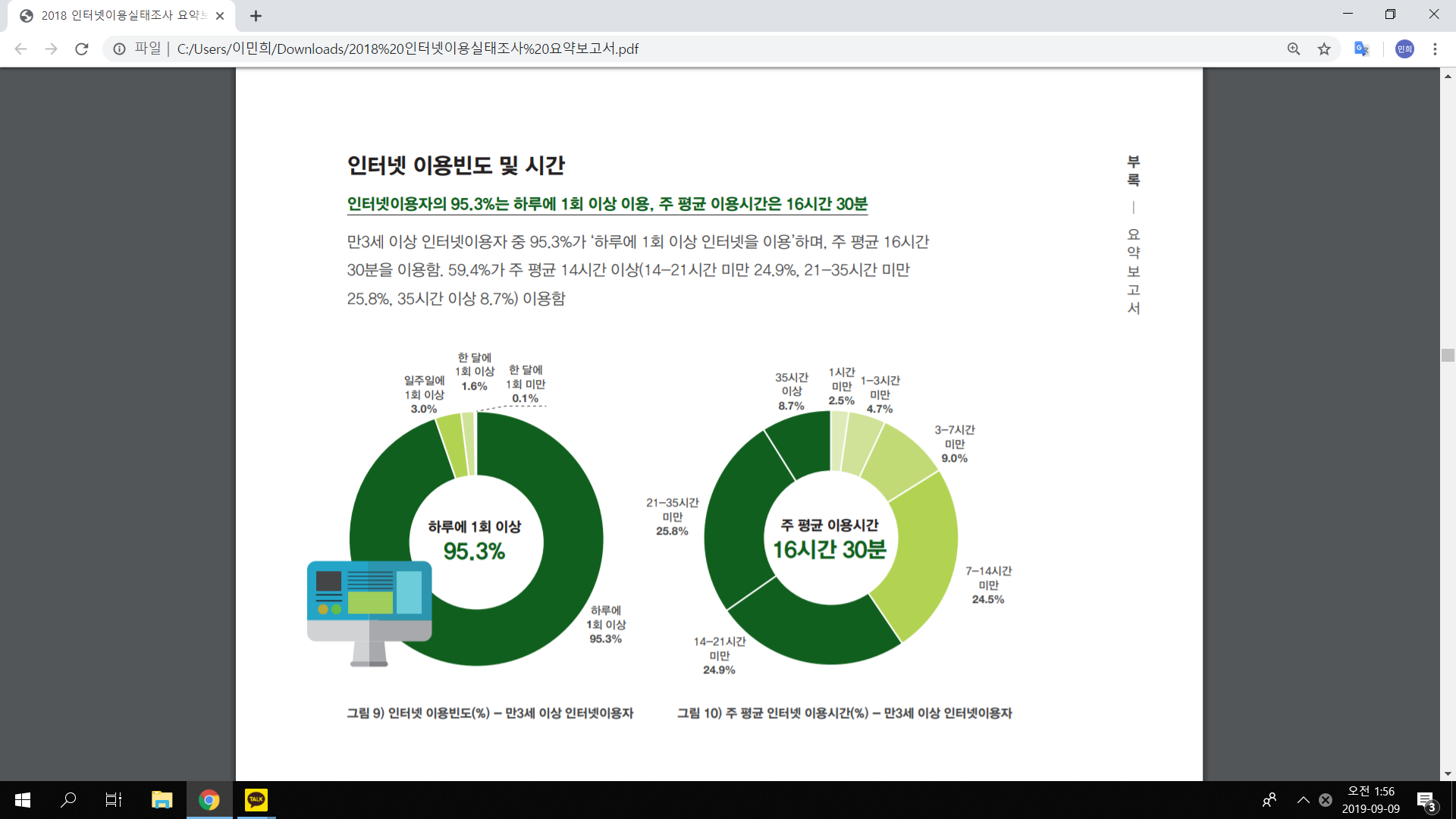
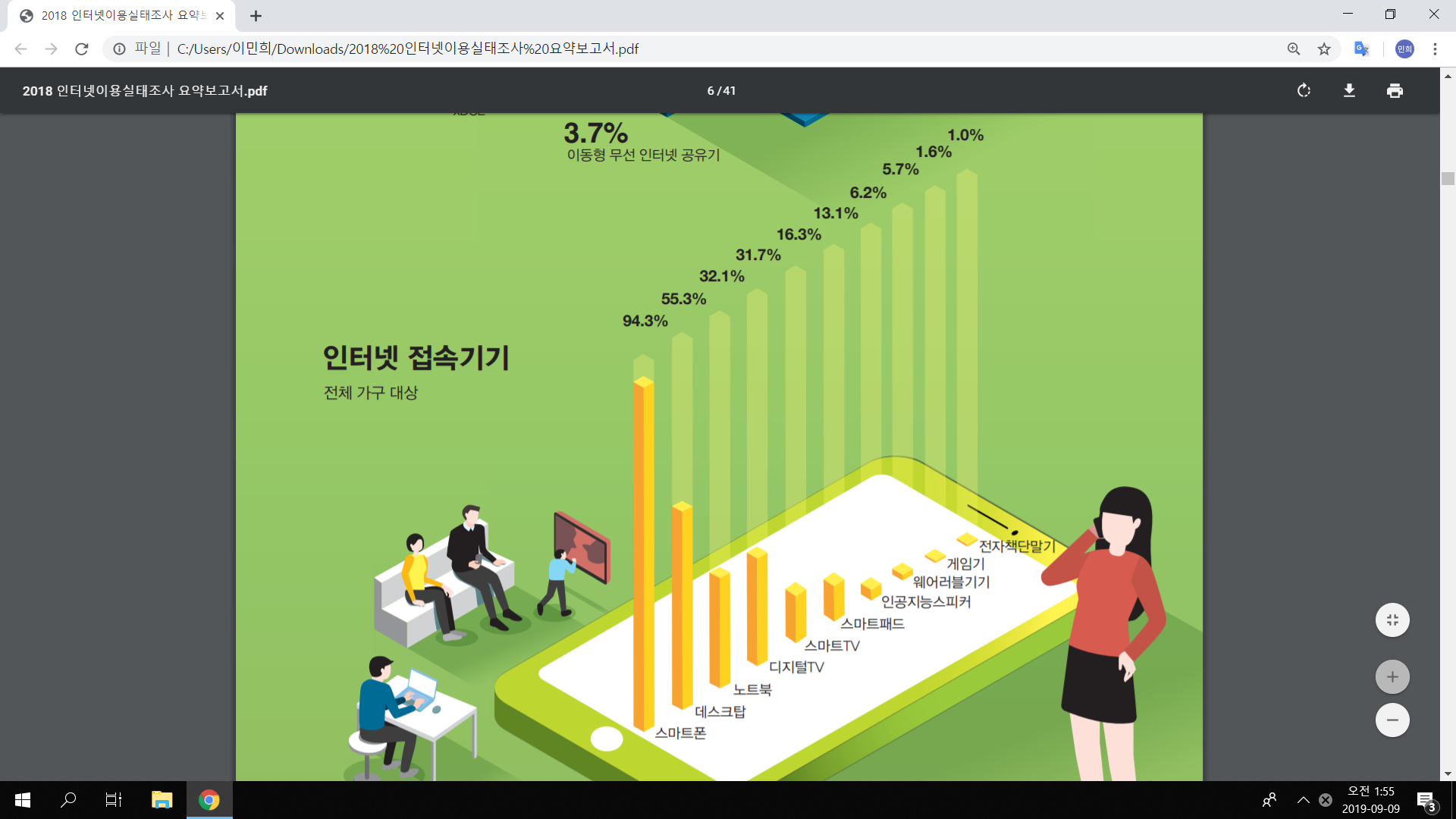


그림 2. 2018 인터넷이용실태조사, 과학기술정보통신부

위 결과로 미루어 보아 웹이나 프로그램 대신 스마트폰 애플리케이션으로 우리의 서비스를 제공하였을 때 사용자가 편하고 쉽게 접근할 수 있을 뿐만 아니라, 사용자와 근접하게 있는 시간이 가장 길다는 점이 별도의 하드웨어 없이도 사용자의 상태 정보를 쉽게 모을 수 있어 본 프로젝트의 목적에 적합하다고 판단하였다.

**- 안드로이드 플랫폼에서 Java, Kotlin 사용**

공식 언어인 Kotlin 발표 이전까지 안드로이드는 Java로 개발을 했기 때문에, Java로 쓰여진 많은 자료(오픈소스 등)를 쉽게 접할 수 있다. 또한 Kotlin은 안드로이드 개발에 적합하고, 여러 기능들을 짧은 코드를 통해 간결하고 알기 쉽게 만들어준다. 이는 우리의 프로젝트의 유지보수가 쉽도록 도와줄 것이다. 따라서 Java와 Kotlin 둘 다 사용하여 서비스를 제작하고자 한다.

**4.2 개발 환경**

* Java 10.0.2, Kotlin 1.3.50
* Android Studio 3.3
* Android SDK Tools 26.1.1
* Android Platform Version API 28 revision 6
* Python3, Tensorflow

**4.3 구현 사항**

**- 기본적인 알람 기능**

기존에 사용하던 알람 App을 이 프로젝트 App으로 쉽게 대체할 수 있도록 기존 알람 App에 탑재 되어있는 기본 기능을 다 구현한다.

가. 설정한 알람 목록 확인 기능

나. 새로운 알람 설정/수정/삭제 기능

다. 알람 상세 정보 설정 (알람 노래 선택, 반복 설정, 상세 정보 기입) 기능

**- 사용자 수면 상태 분석**

이 프로젝트에서 기존의 알람 App과 차별성을 둔 기능은 다음과 같다. 사용자가 설정한 알람을 끈 후 일정시간 내에 스마트폰 센서를 사용해 지속적으로 베개 혹은 침대의 떨림, 외부의 소리가 섞인 숨소리, 스마트폰 사용 등의 데이터를 획득한다. 위 데이터를 Machine Learning을 통해 사용자가 잠이 든 상태인지, 깨어 있는 상태인지를 인식하고, 다시 잠이 들었다고 판단되면 다시 알람을 통해 사용자를 깨워준다.

위의 기능을 구현하기 위해서는 미리 학습이 된 Classification Model이 필요하다. 미리 수집해 전처리한 여러 데이터셋을 잠이 든 상태, 깨어 있는 상태로 구분을 하여 라벨링을 하고, Tensorflow라이브러리를 사용해 데이터 셋들을 라벨링에 맞게 구분하도록 학습시켜 모델을 얻는다. 프로젝트 기간 동안 데이터 셋 변경, 전처리, 모델 구조 변경 등의 활동을 통해 여러 모델들을 생성해가며 구분 정확도를 높인다. 프로젝트 마지막에 가장 높은 정확도를 보이는 모델을 자체 서버 또는 안드로이드 Asset 내에 두고 사용해, 사용자의 새로운 상태 데이터가 들어왔을 때 어떤 상태인지 판단하게 된다.

가. 데이터 셋 수집

나. 데이터 셋 전처리 과정 (숨소리 데이터 싱크 맞추기, 생활 소음과 숨소리 구별, 거리 별 소리 크기 통일 혹은 다양화 등)

다. 모델 설정

라. App에서 인지한 사용자 상태에 따른 반응 구현

마. 사용자 설정(서비스 사용 유무 선택, 몇 분 이내 기능 활성화 선택 등) 기능 추가

**4.4 프로젝트 진행의 어려움**

**- 데이터 셋 확보**

인공지능을 사용한 프로젝트의 대부분 가장 큰 문제는 학습시킬 데이터 셋을 구하기 어렵다는 것이다. 현재 우리도 발견한 수면 상태 혹은 비수면 상태의 데이터셋을 찾지 못해, 데이터 셋을 구하는 것이 첫 어려움이 될 것으로 예상된다. 여러 사이트를 통해 데이터 셋을 찾거나, 직접 데이터셋을 수집하거나, 실제 어플 사용자의 데이터를 수집해 서비스를 제공하는 등의 방향을 생각하고 있다. 데이터 셋의 수가 적어 모델 설계에 문제가 생긴다면 소음 추가, 길이 변경 등의 음성 왜곡 전처리에 시간을 투자해야 할 것으로 보인다.

**- 스마트폰 측정 한계**

이 프로젝트는 사용자가 별도의 하드웨어 장치 구입없이 App만 설치하면 쉽고 간편하게 사용할 수 있도록 스마트폰 센서만으로 사용자의 수면 상태를 인지한다. 따라서 사용자의 심박수, 호흡, 움직임과 같이 직접 접촉해서 얻는 데이터가 아닌, 베개 혹은 침대의 떨림, 외부의 소리가 섞인 숨소리, 스마트폰 사용 등의 사용자 외부의 데이터를 사용해 측정, 인식할 것이다. 이러한 외부 데이터들은 사용자와 스마트폰의 거리, 주변 환경과 같은 외부 요인들의 영향이 크다.

기존에 있는 수면 상태 분석 어플 또한 원리가 정확히 서술되어 있지 않지만 위와 같은 데이터를 사용하였을 것이라고 판단하였다. 실제로 어플이 판단한 상태와 사용자의 체감 상태의 일치가 낮은 정확도를 보이는 것이 외부 데이터들로 정확히 사용자의 상태를 인지하기 어려운 것을 뒷받침한다.

하지만 기존의 어플들은 깨어 있는 상태, 깊은 잠 상태, 얕은 잠 상태 세가지로 상태를 나눠 보여주고, 정리하는 서비스에 초점을 두었으나, 이 프로젝트는 위와 같은 분석 대신 잠이 든 상태인지, 깨어 있는 상태인지로만 인식해 사용자가 잠에서 깬 뒤 다시 잠이 들었을 때 깨워주는 서비스라 좀 더 쉽게 정확도를 올릴 것이라 예상한다. 정확도를 올리기 위해 데이터 전처리, 다양한 측정, 특징 등의 여러가지 방법을 시도해보는 것이 이 프로젝트의 주 챌린지이다.

1. **업무 분담과 프로젝트 스케줄**

**5.1 업무 분담**

* 박시현
* 안드로이드 애플리케이션 GUI 제작
* 수면 데이터 샘플 획득 및 데이터 전처리
* 박진영
* 기본 알람 기능 구현
* 데이터 모델 빌드 및 학습
* 이민희
* 데이터 모델 빌드 및 학습
* 애플리케이션에 모델 적용 및 관련 기능 구현

**5.2 개발 일정**

(공통: 하늘색 / 박시현: 분홍색 / 박진영: 연두색/ 이민희: 보라색 )

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 9월 | | | | 10월 | | | | | 11월 | | | | 12월 | | |
| 주차 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
| 제안서 제출 |  |  |  |  |  |  |  | 중  간  고  사 | 중  간  데  모 |  |  |  |  | 최  종  데  모 | 최  종  리  포  트  제  출 | 기  말  고  사 |
| 안드로이드 GUI 제작 |  |  |  |  |  |  |  |  |  |  |  |
| 기본 알람 기능 구현 |  |  |  |  |  |  |  |  |  |  |  |
| 수면 데이터 샘플 획득 |  |  |  |  |  |  |  |  |  |  |  |
| 데이터 전처리 |  |  |  |  |  |  |  |  |  |  |  |
| 모델 빌드 및 학습 |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
| 어플에 모델 적용 |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
| 모델 관련 기능 구현 |  |  |  |  |  |  |  |  |  |  |  |
| 중간 데모 준비 |  |  |  |  |  |  |  |  |  |  |  |
| 추가 기능 구현 |  |  |  |  |  |  |  |  |  |  |  |
| 테스트 및 디버깅 |  |  |  |  |  |  |  |  |  |  |  |
| 최종 데모 준비 |  |  |  |  |  |  |  |  |  |  |  |
| 최종 보고서 준비 |  |  |  |  |  |  |  |  |  |  |  |