

A Design and Implementation of Health Schedule Application

Ji Woo Kim*, Young Min Lee*, Won Joo Lee*

*Student, Dept. of Computer Science & Engineering, InHa Technical College, Incheon, Korea

*Student, Dept. of Computer Science & Engineering, InHa Technical College, Incheon, Korea

*Professor, Dept. of Computer Science & Engineering, InHa Technical College, Incheon, Korea

[Abstract]

In this paper, we design and implement the HealthSchedule app, which records exercise data based on the GPS sensor embedded in smartphones. This app utilizes the smartphone's GPS sensor to collect real-time location information of the user and displays the movement path to the designated destination. It records the user's actual path using latitude and longitude coordinates. Users register exercise activities and destination points when scheduling, and initiate the exercise. When measuring the current location, a lime green departure marker is generated, and the movement path is displayed in blue, with the destination marker and a surrounding 25-meter radius circle shown in sky blue. Using the coordinates of the starting point or the previous location and the current GPS sensor-transmitted location coordinates, it measures the distance traveled, time taken, and calculates the speed. Furthermore, it accumulates measurement data to provide information on the total distance traveled, movement path, and overall average speed. Even when reaching the destination during exercise, the movement path continues to accumulate until the completion button is clicked. The completion button is activated when the user moves into the sky blue circular area with a radius of 25 meters, centered around the initially set destination. This means that the user must reach the designated destination, and if they wish to continue exercising without clicking the completion button, they can do so. Depending on the selected exercise type, the app displays the calories burned, aiming to increase user engagement and a sense of accomplishment.

▶ **Key words:** Android Studio, GPS Sensor, Geocoder, Naver Map API, Naver Cloude Platform

-
- First Author: Ji Woo Kim, Corresponding Author: Won Joo Lee
 - Ji Woo Kim (sdl13862@naver.com), Dept. of Computer Science & Engineering, InHa Technical College
 - Young Min Lee (leeym008@naver.com), Dept. of Computer Science & Engineering, InHa Technical College
 - Won Joo Lee (wonjoo2@inhac.ac.kr), Dept. of Computer Science & Engineering, InHa Technical College
 - Received: 2024. 01. 02, Revised: 2024. 01. 26, Accepted: 2024. 01. 29.

[요 약]

본 논문에서는 스마트폰에 내장된 GPS 센서를 기반으로 한 운동 데이터를 기록하는 HealthSchedule 앱을 설계하고 구현한다. 이 앱은 스마트폰의 GPS 센서를 활용해 사용자의 실시간 위치 정보를 수집하고 설정한 도착 지점까지의 이동 경로를 보여준다. 위도와 경도 좌표를 사용해 사용자의 실제 경로를 기록한다. 사용자는 스케줄 등록 시 운동 종목 및 도착 지점 등을 등록하고 운동을 시작한다. 현재 위치를 측정하면 연두색 출발지 마커가 생성되고, 이동 경로는 파란색으로 표시되며 도착지 마커와 주위 반경 25M 원은 하늘색으로 표시한다. 출발 지점 또는 직전 위치 좌표와 현재 GPS 센서가 전송한 위치 좌표를 바탕으로 두 좌표 간의 이동 거리, 소요 시간 등을 측정해 속도를 계산한다. 또한, 측정 데이터를 누적하여 전체 이동 거리, 이동 경로 및 전체 평균속도를 확인할 수 있도록 한다. 운동 중 도착 지점에 도달해도 이동 경로는 완주 버튼을 클릭하기 전까지 계속 누적되며 완주 버튼은 초기에 설정한 도착 지점을 기준으로 반지름 25M의 하늘색 원형 안으로 이동하여 하늘색 마커가 생성될 때 활성화된다. 즉 사용자가 설정한 도착 지점에는 반드시 도착해야 하고 추가적인 운동 측정을 원하면 완주 버튼을 클릭하지 않고 운동을 계속할 수 있도록 구현한다. 선택한 운동 유형에 따라 칼로리 소모량이 표시되고, 사용자의 운동 참여를 높여 성취감을 느낄 수 있도록 구현한다.

▶ **주제어:** Android Studio, GPS Sensor, Geocoder, Naver Map API, Naver Cloude Platform

I. Introduction

최근 건강과 신체 활동에 대한 관심이 매우 높아지고 있다. 신체 활동은 개인의 건강에 직접적인 영향을 미치는 중요한 요소 중 하나이다. 신체 활동의 중요성은 주로 건강에 초점이 맞춰져 있지만, 정신적인 면과도 관련이 깊다. 규칙적인 운동은 두통, 스트레스, 변비, 관절염, 불면증, 소화기 장애 등에 뚜렷한 효능이 있으며 특히 심장 질환에 효과를 가져다준다[1]. 또한, 신체 활동은 다양한 질병의 예방과 치료에도 큰 역할을 한다. 코로나19 이후 신체 활동은 개선되었으나 비만, 고혈압, 당뇨병, 고혈압, 심혈관 질환 등과 같은 만성 질환 비율은 증가하였다[2]. 이러한 질환의 발생과 진행을 줄일 수 있는 효과적인 신체 활동이 필요하다. 운동은 단순 미용 목적이나 체중 관리를 넘어, 우리 삶의 질을 높이기 때문에 일반인들의 관심이 지속해서 증가하고 있다.

표 1의 통계청 「사회조사」에 따르면 성별 및 연령집단 별 신체 활동 실천율은 연도별로 꾸준한 상승하는 것을 볼 수 있다[3]. 2018년 기준으로 신체 활동을 꾸준히 실천하는 비율은 38.3%였으며, 이 수치가 2020년에는 40.9%로 상승하였고, 2022년에는 45.5%까지 더 높아진 것을 볼 수 있다. 이러한 결과는 우리 사회가 건강한 신체 활동을 촉진하는 데 관심을 기울이고 있다는 것을 보여주고 있으며, 이것을 통해 미래의 건강한 세대를 양성하고 있는 증거로 볼 수 있다.

운동 데이터를 효과적으로 측정하고 관리하며 수치화하는 것은 중요하기 때문에 정확성과 신뢰성을 가진 데이터 수집 방법 및 도구가 필요하다. 글로벌 헬스 기술 분야의 선도 기업인 로열 필립스(NYSE:PHG·AEX:PIA)가 전문 리서치 업체 칸타 프로파일 네트워크(Kantar Profiles Network)와 함께 발표한 ‘아시아 국가 개인 건강관리 실태 조사’ 자료에 따르면 싱가포르, 태국, 인도네시아 4개국 4,000명을 대상으로 실시한 설문 결과 코로나19 이후 한 국민은 55% 이상이 체온계, 스마트워치, 건강 및 웰빙 앱, 혈압 측정기, 산소 포화도 측정기, 포도당 혈당 측정기 등 건강관리 기기에 관심이 있으며, 이러한 기기를 사용하여 개인의 건강 상태를 모니터링하기를 희망하고 있다[4].

현재 보편적인 기기 중의 하나인 스마트폰을 기반으로 한 운동 관련 애플리케이션이 많이 출시되고 있다. 이러한 앱들은 개인이 쉽게 접근할 수 있고, 운동 활동을 쉽게 기

Table 1. Physical Activity Participation Rate by Gender and Age Group

| | 2018 | 2020 | 2022 |
|--------------------------|------|------|------|
| Entire | 38.3 | 40.9 | 45.5 |
| Male | 40.7 | 43.8 | 47.3 |
| Female | 35.9 | 38.2 | 43.7 |
| Under the age of 20 | 37.1 | 43.4 | 40.6 |
| 20 to 29 years old | 36.8 | 43.8 | 45.7 |
| 30 to 39 years old | 33.2 | 34.3 | 41.1 |
| 40 to 49 years old | 35.7 | 36.4 | 42.0 |
| 50 to 59 years old | 40.8 | 41.2 | 46.7 |
| 60 years of age or older | 43.3 | 45.6 | 50.5 |

록하며 분석할 수 있는 기능을 제공한다. 하지만, 애플리케이션을 사용할 때 전문적인 강사와 함께 운동하는 것이 아니기 때문에 사용자에게 부상의 위험이 존재한다는 문제점이 있다. 이러한 문제점을 줄이기 위해 본 논문에서는 스마트폰 기반의 운동 측정 애플리케이션을 설계하고 구현한다. 이 앱은 사용자가 운동 일정 및 운동 도착 위치를 등록해 주면 자동으로 사용자의 경로, 칼로리 소모량, 이동 거리를 측정해 주기 때문에 운동 측정 데이터를 더 효과적으로 기록하고 수집한다.

본 논문은 다음과 같이 구성한다. 2장에서 기존의 유사 운동 애플리케이션을 분석하고, 애플리케이션의 특징에 대하여 설명한다. 3장과 4장에서는 각각 GPS 센서를 기반으로 사용자의 신체 활동을 기록하는 애플리케이션을 설계하고 구현한다. 마지막으로 5장에서 결론을 맺는다.

II. Preliminaries

1. Analysis of Movement Route Applications

기존 운동 경로 기록 애플리케이션을 분석한 결과는 표 2과 같다. 표 2의 헬스 관련 이동 경로 애플리케이션은 하나의 필수기능과 부가 기능들로 구성되어 있다. 필수기능은 GPS를 사용한 이동 경로 표시로 사용자의 움직임에 따라 실시간 위치를 정확하게 보여준다. 부가적인 기능은 사용자의 성취감을 고취하는 기능을 포함한다. 운동 중 수집한 데이터를 활용하여 사용자의 운동능력을 측정해 준다. 또 휴식 중 사진을 촬영하여 이동 경로에 같이 표시하거나 이렇게 기록한 이동 경로를 다른 사람에게 공유하는 등 여러 방법을 통해 사용자가 다른 사용자와 함께 참여하는 운동으로 유도함으로써 꾸준히 운동할 수 있도록 도움을 준다.

Table 2. Timeline On Map Program

| Classification | My route | My Location Timeline on Map | Geo Tracker |
|----------------|----------|--|------------------------------------|
| Sensor Type | GPS | GPS | GPS |
| Permission | GPS | GPS My Location Timeline on Map | GPS Notifications Background |
| Device | Mobile | Mobile | Mobile |
| Exercise Type | Spinner | X | X |
| Altitude | 0 | X | 0 |
| Speed | 0 | X | 0 |
| Voice Record | 0 | X | X |
| History Share | Global | X | SNS |
| Photo | 0 | X | X |
| Chart | X | X | 0 |

표 2의 앱들은 전 세계 사용자들을 대상으로 하기 때문에 GoogleMap을 사용하고 있다. 하지만 국내에서는 NaverMap이 더 효과적일 수 있다.

2. GPS Sensor

GPS(Global Positioning System)는 1990년 초부터 실제 사용되기 시작한 인공위성을 응용한 센서 모듈이다 [5-7]. 오래된 역사가 있고 현대화와 대중화되었으며 모바일에도 사용된 지 10년이 넘은 만큼 높은 속도와 정확도가 검증되었다.



Fig. 1. GPS Module

3. NaverMap

NaverMap은 Naver Cloud Platform에서 지원하는 서비스 중 하나인 AI·Naver API에서 Application을 등록하고 사용할 수 있다[8]. 애플리케이션 등록 시 안드로이드 스튜디오에서 개발할 앱 패키지 및 사용할 NaverMap의 종류 등을 선택하면 고유 API Key Value가 발행되며 이 키를 사용하여 개발한다.

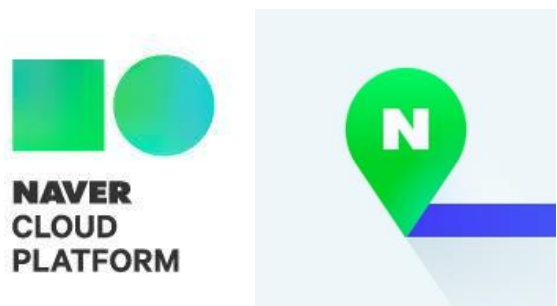


Fig. 2. Naver Cloud Platform

4. Geocoder

Google에서 지원하는 API 서비스인 Geocoder는 위도, 경도 값을 텍스트 주소로 변환해 준다[9-11]. 화면에 표시되는 NaverMap의 정중앙 위도 경도값을 받아와 Geocoder를 사용해 우리나라 도로명 주소로 변환하여 사용한다.



Fig. 3. Google API Service

III. The Design of a HealthSchedule App

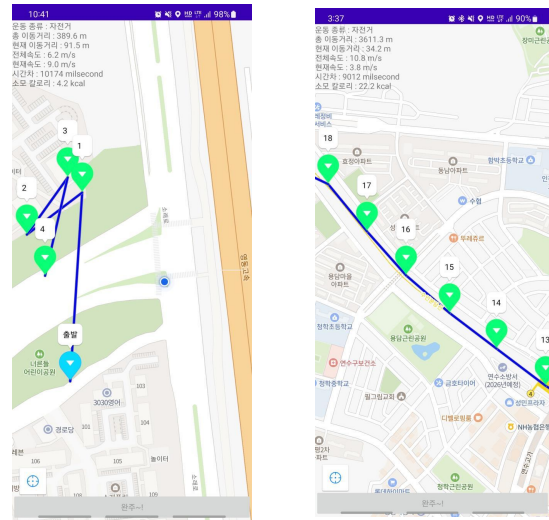
본 논문에서는 NaverMap을 사용해 사용자의 실시간 이동 경로를 보여주며 각 좌표 위치 간의 차이를 계산해 현재 속도 및 이동 거리, 칼로리 소모량 등의 수치들을 보여주는 HealthSchedule 앱을 구현한다[12]. 운동 스케줄 등록 시에는 운동 종목과 도착 위치 등을 설정하며 출발과 도착지에는 파란색 마커로 표시하며 이동 중인 경로는 연두색 마커로 표시한다. 도착지 주변 원을 설정해 사용자의 위치가 원안에 속한다고 판단되면 도착 버튼이 활성화되며 추가적인 경로 측정을 원할 시 운동을 계속할 수 있도록 한다. 또한, 성공적으로 종료한 운동 이력으로 볼 수 있도록 함으로써 성취감을 느낄 수 있도록 설계한다.



Fig. 4. HealthSchedule Application Architecture

1. Position Coordinate Measurement Method

본 논문에서는 GPS 센서를 기반으로 사용자의 이동 경로를 피드백하는 안드로이드 애플리케이션인 HealthSchedule 앱을 구현한다. 사용자의 위치를 측정하는 방법에는 기지국의 신호로 측정하는 방법, 모바일 기기의 GPS로 측정하는 방법 2가지가 있다. 방법마다 위치정확도가 어느 정도인지 확인하기 위해 테스트를 해본다.



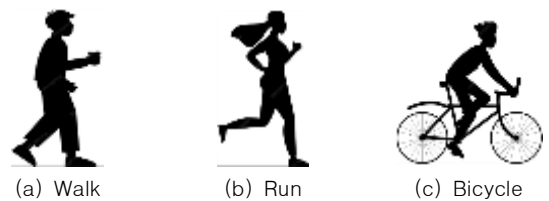
(a) Network (b) GPS

Fig. 5. Error depending on measurement method.

그림 5의 (a)는 기지국으로 측정하는 경우이며 실내에서도 측정할 수 있지만, 오차범위가 크다. 그림 5의 (b)는 GPS 센서로 측정한 경우이며 실내에서 위치를 잡지 못하지만, 실외에서는 높은 정확도를 보여준다. 따라서 실내에서는 이동 경로를 측정할 만한 운동이 적으며 측정한다고 해도 오차가 크기 때문에 GPS 센서를 사용하여 애플리케이션을 설계한다.

2. Sports Type

HealthSchedule 앱의 운동 종목은 그림 6처럼 걷기, 달리기, 자전거로 분류한다. 분류하며 Spinner로 구현해 추후 종목 추가를 쉽게 하도록 구현한다. 또 종목별 칼로리 소모량을 계산한다.



(a) Walk (b) Run (c) Bicycle

Fig. 6. Type of exercises

IV. The Implementation of a HealthSchedule App

본 논문에서 구현하는 안드로이드 애플리케이션은 스마트폰의 GPS 센서에서 측정한 위치 수치를 사용하여

NaverMap에 현재 위치를 표시한다. HealthSchedule 앱 구현에 필요한 클래스는 표 3과 같다.

Table 3. Function of Class

| Class | Function and Method |
|-------------------------|--|
| ScheduleList Activity | Exercise Schedule List • showScheduleList() • btnExerciseStart() • btnScheduleUpload() • clearSelection() • DeleteSelectedItems() |
| ScheduleUpload Activity | Exercise Schedule Registration • btnRegisterNaverMap() • btnRegisterClick() • onActivityResult() • onChanged() • onRequestPermissionsResult() |
| ScheduleUpload NaverMap | Set Arrival Location • onMapReady() • mDisplayMaker() • addOnCameraChangeListener() • addOnCameraIdleListener() • onRequestPermissionsResult() • getDistance • getAddress() |
| NaverMap | Display User Movement Path • onMapReady() • mDisplayMaker() • mUpdateMap() • onRequestPermissionsResult() • getDistance() • getAddress() • onBackPressed() |
| DBHelper | SQLite Integration • DBHelper() • onCreate() • onUpgrade() • deleteSchedule() • completeScheduleUpd() |

표 3의 클래스들을 순서대로 스케줄 조회, 등록, 운동 시작, 완료 순으로 설명한다.

1. Application Development Environment

본 논문에서 개발하는 HealthSchedule 앱은 Windows 기반의 Android Studio를 사용하여 개발하였으며 실제 테스트한 모바일 기기는 Galaxy S22+ 이다. 자세한 개발 환경은 표 4과 같다.

Table 4. Development Environment

| Classification | Environment |
|----------------|---|
| Hardware | • Galaxy S22+ (Tiramisu) |
| SoftWare | • Java SDK 1.8 • Android Studio Electric Eel 2022. 1. 1 Patch 2 • Android 13.0 (Tiramisu) |

그림 7 (a)는 스케줄 목록 화면에서 스케줄 등록 버튼 클릭 시 btnScheduleUpload() 메서드를 호출해 그림 7

(b)와 같은 등록 화면으로 전환해 주며 운동 소요 시간과 운동 종류, 날짜 및 도착 위치를 선택하도록 구현한다.



Fig. 7. Schedule List & Register Screen

도착 위치 지정 버튼 클릭 시 onRequestPermissionsResult() 메서드로 위치 접근 권한을 확인하여 그림 8의 (a)와 같이 표시해 주며 그림 8의 (b)처럼 NaverMap의 정중앙의 마커를 기준으로 getAddress() 메서드를 호출하여 도착 위치를 설정한다.

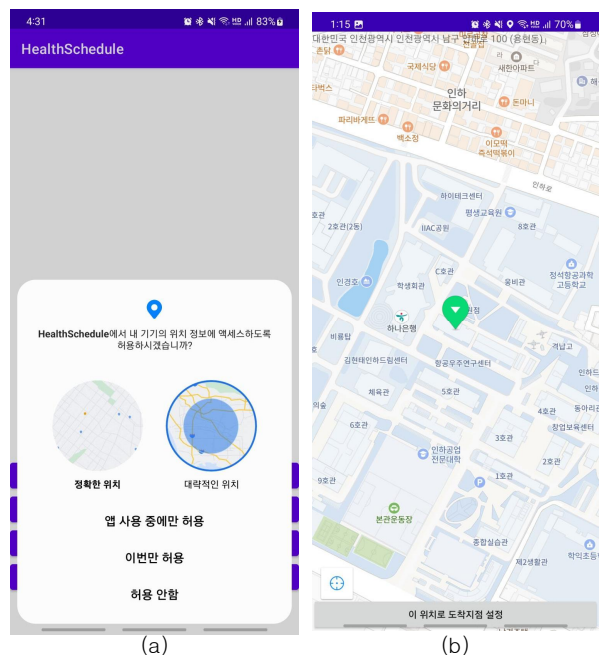


Fig. 8. Location Setting Screen

마커 정중앙 표시는 표 5와 같다.

Table 5. Center Maker Display Code

```
CameraPosition cameraPosition =
mMap.getCameraPosition();
double dLatitude = cameraPosition.target.latitude;
double dLongitude=cameraPosition..target.longitude;
Center_location = new LatLng(dLatitude, dLongitude);
Marker centerMarker = new Marker();
centerMarker.setPosition(Center_location);
centerMarker.setMap(mMap);
centerMarker.setVisible(true);
```

표 6은 도착 위치값을 Geocoder 서비스를 이용하여 좌측 상단에 텍스트 주소를 표시하여 사용자가 쉽게 알아볼 수 있도록 한다. 도착 위치까지 설정하였다면 스케줄 등록 버튼을 클릭하여 등록 후 목록으로 돌아간다.

Table 6. getAddress Method Code

```
String nowAddress = "현재 위치를 확인 할 수 없습니다.";
Geocoder geocoder = new
Geocoder(mapFragment.getContext(), Locale.KOREA);
List<Address> address = geocoder.getFromLocation(lat, lng, 1);
String currentLocationAddress =
address.get(0).getAddressLine(0).toString();
nowAddress = currentLocationAddress;
```

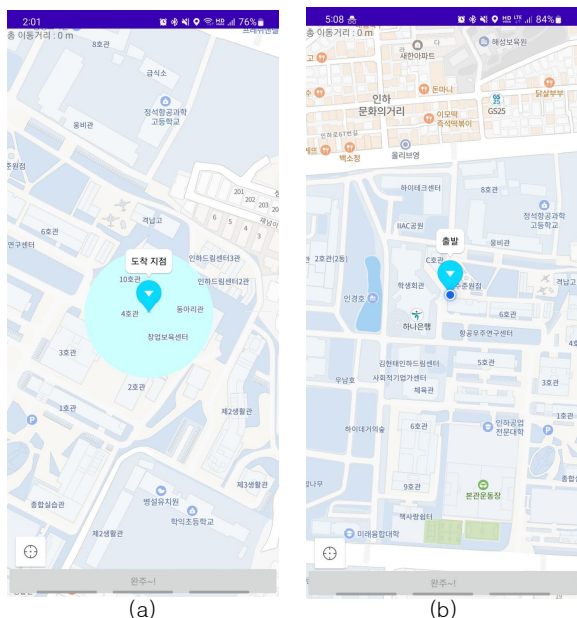


Fig. 9. Start Exercise

등록이 완료되었으면 원하는 스케줄 선택 후 선택한 스케줄로 운동 시작 버튼을 클릭 시 그림 9의 (a)와 같이 파란색 마커와 지름 50m의 원이 표시되며 도착 지점을 알려 준다. 이후 현재 위치가 감지되면 그림 9의 (a)와 같이 파

란색으로 출발 마커가 표시되며 필요시에 좌측 하단의 아이콘 클릭 시 더 빠르게 현재 위치를 찾을 수 있다.

표 7은 출발지, 도착지, 이동 경로 출력에 사용하는 마커 표시 mDisplayMaker() 메서드이다.

Table 7. mDisplayMaker Method Code

```
mDisplayMarker(LatLng objLocation, String color) {}
Marker objMK = new Marker();
objMK.setVisible(false);
objMK.setPosition(objLocation);
objMK.setIconTintColor(Color.parseColor(color));
objMK.setMap(mMap);
objMK.setVisible(true);
```

그림 10의 (a)처럼 출발 후 10초 또는 20m마다 마커가 생성되며 이동 경로를 기록한다. 각 마커가 생성되면 직전 마커와 현재 마커의 거리 및 소요 시간을 계산하여 전체 이동 거리, 현재 속도 및 평균속도, 운동 종목별 칼로리 소모량 등을 계산하여 보여준다. 도착 지점에 근접하면 파란색 마커로 색이 변경되며 완주 버튼이 활성화된다. 완주 버튼 클릭 시 계산한 수치들이 저장되며 추가적인 경로 측정을 원할 때 도착 지점을 경유하면 버튼은 활성화된 상태가 유지된다.

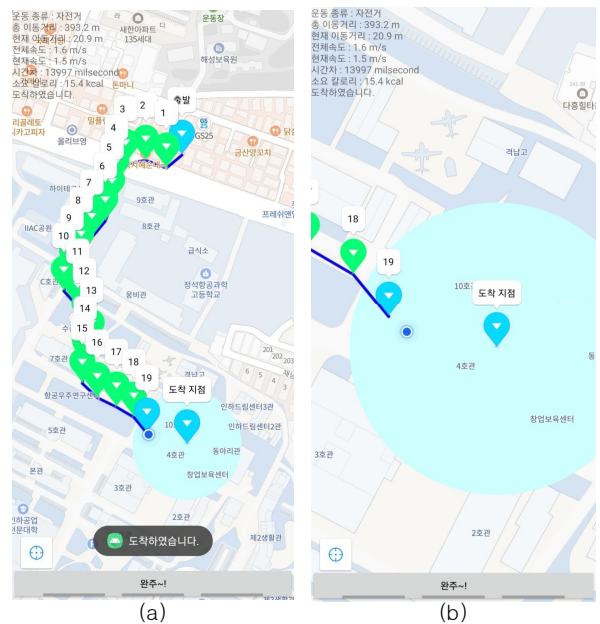


Fig. 10. End Exercise

표 8은 두 마커 간의 위도 경도 차이를 m 단위로 직선 거리를 계산하는 코드이다. 그림 10의 (a) 좌측 상단에 부가 정보 중 전체 및 직전 마커와의 이동 거리를 계산해 표시해 준다.

Table 8. getDistance Method Code

```

getDistance(double lat1, double lon1, double lat2,
double lon2)
    double R = 6372.8 * 1000;
    double dLat = Math.toRadians(lat2 - lat1);
    double dLon = Math.toRadians(lon2 - lon1);
    double a = Math.pow(sin(dLat / 2), 2) +
    Math.pow(sin(dLon / 2), 2) *
    cos(Math.toRadians(lat1)) * cos(Math.toRadians(lat2));
    double c = 2 * asin(sqrt(a));
    return (R * c);

```

표 9는 마커가 표시된 두 좌표 간 이동 거리를 활용하여 현재 속도, 전체 속도를 계산한다.

Table 9. Arrival Detection Code

```

if (Math.abs(arrivedLatitude - Current_location.latitude)
<= 0.0005 && Math.abs(arrivedLongitude -
Current_location.longitude) <= 0.0005) {
    double R = 6372.8 * 1000;
    mDisplayMarker(Current_location, "#0000FF");
    Toast.makeText(getApplicationContext(),
    "도착하였습니다.", Toast.LENGTH_SHORT).show();
    totalDistanceTextView.append("도착하였습니다.\n");
} else {
    mDisplayMarker(Current_location, "#00FF00");
}

```

또한, 그림 10의 (b)와 같이 도착 지점 지름 50m 안에 마커가 위치하였는지 판별하고, 도착 지점 판별 시 도착 버튼을 활성화하며 마커를 하늘색으로 표현해 준다.

표 10은 표 8에서 기록한 마커 간 시간대 편차를 활용하여 스케줄마다 선택한 운동 종목의 시간당 평균 칼로리 소모량을 계산하여 보여주는 코드이다. 이 계산식은 운동 중 잠시 휴식 및 격한 운동 등에 따른 오차는 상정하지 않는다.

Table 10. Calculation Calorie Code

```

if ("걷기".equals(exerciseGubun)) {
    costCaloriePerSecond = 2.6 / 60.0;
} else if ("달리기".equals(exerciseGubun)) {
    costCaloriePerSecond = 11.6 / 60.0;
} else if ("자전거".equals(exerciseGubun)) {
    costCaloriePerSecond = 4.0 / 60.0;
}
totalCostCalorie = (currentTime - startTime) / 1000.0 *
costCaloriePerSecond;

```

이 논문에서는 스케줄 등록과 운동 내용 확인에 SQLite DB를 사용한다. 표 11은 스케줄 등록 시 DB에 등록된 정보를 삽입하기 위한 메서드다.

Table 11. onCreate Method Code

```

public void onCreate(SQLiteDatabase db) {
    String createTableQuery =
    "CREATE TABLE schedule ("
    + (스케줄 등록시 필요한 컬럼 나열) + ")";
    db.execSQL(createTableQuery);
}

```

표 12는 사용자가 선택한 스케줄을 DB에서 삭제하는 메서드이다.

Table 12. deleteSchedule Method Code

```

public void deleteSchedule(int hour, int year, int month,
int day, String arrived_address, double arrived_latitude,
double arrived_longitude) {
    SQLiteDatabase db = getWritableDatabase();
    month--;
    db.execSQL("DELETE FROM schedule WHERE
healthhour = " + hour + " AND year = " + year
+ " AND month = " + month + " AND day = " + day
+ " AND arrived_address = " + arrived_address + ""
+ " AND arrived_latitude = " + arrived_latitude
+ " AND arrived_longitude = " + arrived_longitude);
    db.close();
}

```

V. Conclusions

본 논문에서는 스마트폰에 내장된 GPS 센서를 기반으로 한 운동 데이터를 기록하는 HealthSchedule 앱을 설계하고 구현하였다. 이 애플리케이션은 사용자의 운동을 다양한 측면에서 기록하고 분석할 수 있는 다양한 기능을 제공한다. 사용자는 운동 종목과 도착지, 소요 시간 및 날짜를 등록하여 운동을 시작한다. 이러한 정보 등록을 통하여 운동 활동을 문서화하고, 사용자의 운동 스케줄을 체계적으로 관리한다. 이 애플리케이션은 GPS 센서를 활용하여 사용자의 현재 위치를 위도와 경도 값을 통해 실시간으로 전송받는다. 이 정보를 NaverMap과 연동하여 사용자의 이동 경로를 추적한다. 이를 통해 사용자는 운동 중 어느 경로로 이동했는지 시각적으로 확인할 수 있다. 또한, 사용자가 목표한 도착 지점에 도달하면 운동 완료 여부가 표시되어 사용자는 목표를 달성하는 데 필요한 동기부여를 얻을 수 있다. 운동 중 사용자에게 현재 속도, 이동 경로, 이동 거리와 같은 정보를 실시간으로 제공함으로써, 목표한 지점까지의 진행 상황을 쉽게 파악할 수 있도록 한다. 그뿐만 아니라, 운동을 시작할 때 선택한 운동 종목에 따른 칼로리 소모량도 표시한다. 이처럼 운동 관련 데이터의 기록과 분석을 통해 사용자는 더욱 효과적으로 운동을

계획할 수 있고, 개별적인 목표를 달성하는 데 성취감을 크게 느낄 수 있을 것이다. 이 애플리케이션은 사용자들에게 건강한 라이프 스타일을 유지하는 데 도움을 줄 것으로 기대한다. 향후 연구과제는 HealthSchedule 앱의 성능과 정확도를 향상시키는 것이다.

REFERENCES

- [1] <https://www.ydp.go.kr/health/contents.do?key=3637&>
- [2] <https://www.k-trendynews.com/news/articleView.html?idxno=162615>
- [3] Statistics Korea, "Physical Activity Participation Rate by Gender and Age Group," 2006-2022, <https://www.index.go.kr/unify/idx-info.do?idxCd=5068>
- [4] PHILIPS, "Healthy Living in Asia Survey," 2022, <https://www.philips.co.kr/a-w/about/news/archive/standard/about/news/press/2022/20220720-philips-announces-results-of-personal-health-management-survey-in-asia.html>
- [5] <https://www.fpn119.co.kr/132109>
- [6] W. J. Lee, S. M. Lee, S. H. Han, J. H. Hong, "A Design and Implementation of Disaster Prevention," The Proceeding of Korea Society of Computer and Information 2023 Summer Conference, Vol.30, No.2, pp. 237-238, July 2022
- [7] J. H. Ahn, H. J. Lee, S. Y. Lee, J. W. Han, W. J. Lee, "A Design and Implementation of Local Festivals and Travel Information Service Application," Journal of The Korea Society of Computer and Information (ISSN 1598-849X), Vol. 28, No. 11, pp. 65-71, Nov. 2023, <https://doi.org/10.9708/jksci.2023.28.11.065>
- [8] <https://www.ncloud.com/>
- [9] <https://developer.android.com/reference/android/location/Geocoder>
- [10] <https://developers.google.com/maps/documentation/javascript/geocoding?hl=ko>
- [11] https://www.vworld.kr/dev/v4dv_geocoderguide2_s001.do
- [12] W. J. Lee, J. W. Kim, Y. M. Lee, "A Design and Implementation of Health Schedule Application," The Proceeding of Korea Society of Computer and Information 2023 Summer Conference, Vol. 31, No. 2, pp. 441-442, July 2023

Authors



Ji Woo Kim received a bachelor's degree in professional science in the Smart IT department of Baehwa Women's College in Korea in 2023. The student entered Inha Technical College in 2023 for a bachelor's

degree in Computer and Information Engineering. She is interested in the web and apps.



Young Min Lee received the Associate Degree in Computer Science from Inha Technical College, Korea, in 2023. He received B.S. degree in Computer Science Engineering from Inha Technical College,

Korea, in 2024. He is interested in mobile computing, artificial intelligence, and cloud computing, data science.



Won Joo Lee received the B.S., M.S. and Ph.D. degrees in Computer Science and Engineering from Hanyang University, Korea, in 1989, 1991 and 2004, respectively. Dr. Lee joined the faculty of the Department of

Computer Science at Inha Technical College, Incheon, Korea, in 2008, where he has served as the Director of the Department of Computer Science. He is currently a Professor in the Department of Computer Science, Inha Technical College. He has also served as the Vice-president of The Korean Society of Computer Information. He is interested in parallel computing, internet and mobile computing, and cloud computing, data science, artificial intelligence.