

<http://dx.doi.org/10.7236/IIBC.2014.14.4.141>

IIBC 2014-4-21

스마트 폰을 위한 센서기반 건강 다이어트 어플의 설계

Design of Sensor-based Healthy Diet App for Smartphones

오선진*

Sun-Jin Oh*

요약 최근 스마트 폰 주요 기능의 급속한 발전과 더불어 스마트 폰과 연계된 많은 응용들이 속속 개발되고 있으며, 다양한 관련 기술과의 융합을 통해 획기적인 모바일 응용들이 개발되어 빠르게 확산되고 있는 실정이다. 본 논문에서는 이러한 스마트 폰의 다양한 기능들을 효과적으로 활용할 수 있도록 주변의 센서와 같은 장치들과 연동한 스마트한 모바일 응용의 설계를 제안한다. 본 논문에서 제안한 모바일 응용은 스마트 폰의 블루투스 모듈을 이용하여 주변의 다양한 상황정보 센서들과 일시적인 애드 혹 망을 형성하고 사용자의 생체 상황정보를 실시간으로 수집하여, 사용자가 효과적인 다이어트를 수행하는데 반영하여 건강한 다이어트를 수행할 수 있도록 하는 스마트 폰을 위한 센서 기반 건강 다이어트 어플을 설계하고 구현하였다.

Abstract With the rapid development of major functions in a smartphone, many applications interacting with a smartphone are introduced and mobile applications converged with various related technologies are announced and rapidly disseminated recently. In this paper, we propose the design of a smart mobile application operated with devices like sensors in order to utilize various functions of the smartphone effectively. The mobile application proposed in this paper uses the bluetooth module of the smartphone, composes the wireless ad hoc network with various sensors around us, and collects bionic context information of the user in real time. The smart application designed and implemented in this paper is the sensor based healthy diet application that can fulfill healthy diet program by reflecting the user's sensed bionic context data in real time.

Key Words : Bluetooth, Healthy Diet App, USN

1. 서론

스마트 폰 단말기 제조 기술의 급속한 발전으로, 스마트 폰의 기능과 응용 범위가 매우 다양하게 확장되었고, 여러 다른 분야의 기술들과 활발하게 융합이 이루어지고 있어 최근 스마트 응용의 주류가 되고 있다. 요즘 출시되고 있는 스마트 폰 등 단말기를 보면 다양한 장치들을 탐

재하고 있으며, 최근 출시된 삼성전자의 Galaxy S5 모델의 경우 무려 10개의 각종 센서들을 탑재하고 있어 이를 이용한 다양한 형태의 스마트 응용들의 구현이 가능하게 되었고, 이를 통한 다양한 분야의 기술들과의 융복합 형태로 스마트 응용은 지속적으로 확대 발전되고 있는 상황이다. 또한 최근의 스마트 응용들은 기존의 WiFi 망 뿐만 아니라 다양한 통신 채널들과 결합한 유비쿼터스 센

*종신회원, 세명대학교 정보통신학부
접수일자 : 2014년 6월 20일, 수정완료 : 2014년 7월 13일
게재확정일자 : 2014년 8월 8일

Received: 20 June, 2014 / Revised: 13 July, 2014 /
Accepted: 8 August, 2014

*Corresponding Author: sjoh@semyung.ac.kr

Dept. of Information & Communication Technology, Semyung University, Korea

서 네트워크, ZigBee나 블루투스 기반 애드 혹 센서 네트워크 응용분야에 이르기까지 매우 다양하게 개발되고 있으며 그 발전 가능성은 앞으로도 무궁무진하다고 할 수 있다.^[1] 특히 최근의 스마트 응용들은 컴퓨팅 기능과 통신 기능이 융합한 응용, 스마트 폰 상의 주변기기와 주변의 센서 모듈들과의 연계를 통한 융합 스마트 응용, 공통 취미나 특기, 관심을 가진 그룹을 구성하는 사용자들 간의 소셜 네트워크에 기반한 소셜 관련 스마트 응용, GPS나 3G 지리정보 등을 이용한 공간적, 지형학적 특징을 이용하는 LBS기반 스마트 응용, 개성 중심의 개인 라이프 스타일이나 취향에 맞춘 맞춤형 스마트 응용에 이르기까지 실생활과 관련된 편의성과 효율성을 강조하는 다양한 기능성 스마트 응용들이 주류를 이루고 있다.^[2,3] 아울러 이러한 스마트 응용의 설계와 구현을 보다 쉽고 간편하게 할 수 있도록 도와주는 스마트 응용을 위한 미들웨어에 대한 연구 또한 최근 다양한 개발 플랫폼 환경 하에서 편리하게 사용될 수 있도록 본격화 되고 있는 상황이다.^[4,5]

본 논문에서는 스마트 폰을 위한 센서 기반의 건강 다이어트 어플을 설계하고 구현하였다. 본 논문에서 설계하고 구현한 스마트 응용은 사용자가 다이어트를 스마트 폰을 이용하여 수행할 수 있도록 응용 프로그램을 설계하였다. 다이어트를 수행하는 사용자는 스마트 폰을 이용해 매일 섭취하는 음식의 종류와 양을 입력받아 총 섭취한 칼로리를 계산하고, 이렇게 섭취한 칼로리를 효과적으로 소모하기 위한 효율적인 운동 종류 선택과 처방 및 칼로리 관리를 수행하게 된다. 여기서 사용자의 건강한 다이어트를 유도하기 위하여 현재 수행중인 다이어트 프로그램에 사용자의 현재 건강 생체 정보를 실시간으로 측정하여 반영할 수 있도록 추가로 주변의 생체 데이터 측정 센서와 무선으로 연동하여 건강 다이어트 프로그램에 적용할 수 있도록 하였다. 이를 위해 안드로이드 기반 스마트 폰에 기본 장착되어 있는 블루투스 모듈을 이용하여 주변의 체온, 맥박, 산소포화도 등의 생체 정보를 실시간으로 측정할 수 있는 다중 센서를 무선으로 연결하여 일시적인 무선 애드 혹 센서 네트워크를 구성하고, 다이어트를 위해 운동을 하고 있는 사용자의 맥박과 체온, 산소포화도 등 생체 데이터를 실시간으로 측정하여 전송하게 함으로써 현재의 사용자의 상태 정보를 획득하게 하였다. 이와 같이 획득된 데이터를 데이터베이스에 저장하고 분석하여 사용자의 건강 상태를 결정하고 이 결

과에 따라 적절하게 운동 처방을 하는 센서 기반 건강 다이어트 스마트 응용을 설계하고 구현하였다.

본 논문의 구성은 다음과 같다. 2장에서는 최근 스마트 폰의 첨단 기능과 센서 기술 관련 연구들을 살펴보고, 3장에서는 본 논문에서 설계한 센서기반 건강 다이어트 응용의 시스템 모델을 알아보고, 4장에서는 구성 모듈의 주요 동작과 알고리즘을 서술하였으며, 5장에서는 제안한 센서 기반 건강 다이어트 응용의 모듈별 구현 및 실행 결과를 고찰하였고, 마지막으로, 6장에서 향후 연구과제와 함께 결론을 맺는다.

II. 관련 연구

스마트 폰 단말기 제조 기술의 급속한 발전으로, 스마트 폰의 기능과 응용 범위가 매우 다양하게 확장되고 있다. 최근 출시된 삼성전자의 '갤럭시 S5'에는 첨단 전자 센서(sensor)가 10개 내장돼 있다. 손바닥에 쑥 들어오는 조그만 스마트폰이 이 센서들을 통해 내가 지금 어디서 무얼 하는지, 몸 컨디션은 어떤지 속속들이 파악한다. 세계 최초로 스마트폰에 내장된 심박 센서가 대표적이다. 심박수는 체중·체온·혈압·호흡량 등과 함께 인체의 상태를 알려주는 가장 기본적인 정보다. 과거에는 심박수를 측정하려면 병원에서 습기를 머금은 전극 패드 2장을 몸에 나눠 붙인 다음 측정해야 하는 복잡한 과정을 거쳐야 했다. 하지만 갤럭시 S5를 이용하면 뒷면의 센서에 손가락을 대기만 하면 심박수가 측정된다. 심박 센서는 붉은색 발광다이오드(LED)와 파동 센서로 구성된다. LED가 붉은빛으로 손가락 안쪽의 가느다란 혈관을 비추면, 파동 센서가 혈관 속 적혈구 움직임을 파악해 심박을 측정하는 원리다. 측정에 걸리는 시간도 단지 8초다. 이렇게 측정된 심박수는 스포츠 분야에서 운동 강도를 조절하고 적절한 휴식을 취하는데 활용될 수 있다.^[6]

지문 인식 센서는 갤럭시 S5의 하단 중앙 '홈 버튼'에 내장돼 있다. 손가락으로 홈 버튼을 위에서 아래로 문지르기만 하면 된다. 센서가 사용자의 지문을 읽어 들여 미리 암호화해 저장한 지문 패턴 데이터와 비교해 이 스마트폰의 주인이 맞는지 확인한다. 위치인식 센서 기능도 사용자가 산에 있는지, 바다에 있는지 등을 입체적으로 파악해 적절한 정보를 알려준다. 위성 위치신호인 GPS, 현재 이동 상태를 파악해 만보계 등에 활용할 수 있는

가속도 센서, 스마트폰의 각도와 자세를 파악하는 자이로(gyro) 센서에 더해 기압 센서까지 들어있다. 기압 센서는 대기압(大氣壓)을 측정해 사용자가 있는 곳의 고도를 파악한 후, 사용자의 위치를 정확하게 파악할 수 있도록 한다. 또 계단을 오르거나 등산할 때 소모 칼로리를 더 정확히 계산해낸다.^[6,7]

이외에도 커버가 열렸는지 닫혔는지를 인식해 화면을 자동으로 켜거나 끄는 홀센서, 지구 자력을 측정해 정확한 방향을 알아내는 지자기 센서, 주변 밝기에 맞춰 화면 밝기를 자동 조절하는 RGB 조도센서, 사용 중 얼굴이 가까워지면 액정화면을 자동으로 끄는 근접센서, 터치없이 손동작으로 스마트 폰 기능을 작동하는 제스처 센서 등이 내장되어 있다. 이와 같이 첨단 스마트폰 단말은 무척 빠른 속도로 그 기능이 진화하고 있다.^[7,8]

안드로이드 기반 스마트 기기에서의 센서를 이용하는 어플을 개발하는 것은 안드로이드 기기의 감지 능력에 대한 이해와, 어플에서 어떤 센서를 사용할 것인지에 대한 선택 그리고 센서 데이터를 획득하고 분석할 수 있도록 어플에 적용하는 것들을 포함한다. 안드로이드 기반 스마트 기기는 매우 다양한 종류의 센서들과 사용할 수 있다.^[7,8] 스마트 폰이 등장하기 전에 사용자들은 좁은 영역에서만 센서를 활용할 수 있었다. 보통 하나의 기기에 하나의 센서를 활용하였고, 한가지의 목적으로 사용되도록 고안되었다. 그러나 스마트 폰의 등장은 사용자와 개발자들에게 센서를 활용하는 흥미로운 새로운 영역을 열어주었다. 안드로이드 1.5(API 레벨 3)부터 표준 센서들과 관련된 센서 API들을 사용할 수 있게 되었고, 안드로이드 2.3(API 레벨 9)에서는 Android developer's toolbox에 새로운 센서와 툴들이 추가되었다. 현재 포함되어 있는 표준 센서들에는 가속도계, 자이로스코프, 나침반, 조도 센서, 근접 센서, 상대 습도계, 기압계 등이 있다. API 레벨 9에 추가된 툴로는 회전 행렬과 쿼터니언, 합성된 센서를 얻는 함수들이 있다. 이를 이용해 물리적인 내비게이션, 게임 컨트롤, 증강현실 및 다양한 용도에 응용될 수 있다.^[8,9]

III. 시스템 모델

최근 첨단 스마트 기기 제조 기술의 눈부신 발전과 더불어, 스마트 기기의 급속한 보급과 이를 이용한 스마트

응용 기술의 획기적인 발전으로 많은 사용자들이 스마트폰 응용에 대한 지대한 관심과 많은 참여가 이루어지고 있다. 본 논문에서는 안드로이드 OS기반 스마트 기기를 위한 센서 기반의 건강 다이어트 어플을 설계하고 구현하였다. 본 논문에서 설계하고 구현한 스마트 응용은 사용자가 보다 효과적인 다이어트 수행을 할 수 있도록 스마트폰을 이용하여 매일 섭취하는 음식의 종류와 양을 입력받아 총 일일 섭취한 칼로리를 계산하고, 이렇게 섭취한 칼로리를 효과적으로 소모하기 위한 효율적인 운동 종류 선택과 운동 처방 및 소모 칼로리 관리를 수행하게 된다. 여기서 사용자의 건강한 다이어트를 유도하기 위하여 현재 수행중인 다이어트 프로그램에 체온, 맥박, 산소 포화도 등의 생체 정보를 실시간으로 측정하여 반영할 수 있도록 하였다. 이를 위해 안드로이드 기반 스마트폰에 장착되어 있는 블루투스 모듈을 이용하여 주변의 온도, 습도, 맥박, 산소포화도 등의 생체 정보를 실시간으로 측정할 수 있는 다중 센서를 무선으로 연결하여 무선 애드 혹 센서 네트워크를 구성하고, 다이어트를 위해 운동을 하고 있는 사용자의 맥박과 체온, 산소포화도 등 생체 데이터를 실시간으로 측정하여 전송하게 함으로써 현재 사용자의 생체 상태 정보를 획득하게 하였다. 이와 같이 획득된 데이터를 데이터베이스에 저장하고 분석하여 사용자의 현재 건강 상태를 결정하고 그 결과에 따라 적절하게 운동 처방을 조절할 수 있게 하는 센서 기반 건강 다이어트 스마트 응용을 설계하고 구현하였다. 이 어플은 구글의 안드로이드 OS를 기반으로 하는 모든 스마트 기기에 내장된 블루투스 모듈을 통해 획득된 다중 센서로부터의 실시간 생체 데이터를 저장하고, 이를 이용해 사용자의 건강 상태를 판단하고 의사결정을 하는 헬스케어 스마트 응용 보조 계측장비로 이용될 수 있다.

본 논문에서 제안한 스마트 폰을 위한 센서 기반 건강 다이어트 어플을 위한 시스템 모델은 다음의 표 1에 보인바와 같다. 어플 개발을 위한 소프트웨어 개발 툴로 이클립스 Juno Service Release 1을 사용하였고, 프로그래밍 언어는 자바 JDK 1.7.0_25를 사용하였다. 여기에 안드로이드 API 함수 사용을 위해 Android SDK 4.2 (Jelly Bean)를 사용하였고 자바와 연동하기 위해 Android ADT를 설치하였으며, 구현한 어플의 실행 결과 확인을 위해 AVD 16-wvga.avd 에뮬레이터를 설치하여 사용하였다. 이 어플 개발을 위한 하드웨어 플랫폼으로는 3세대 인텔 코어 i5가 장착된 삼성 노트북에 32bit 윈도우 7 운

영체제 하에서 웹 접속을 위한 Tomcat 7.0과 데이터베이스로 MySQL 5.5를 설치하여 사용하였다. 센서 모듈은 HBX_ZigBeX II mote 기반 모듈에 블루투스 통신 모듈을 장착하고, 체온과 맥박, 산소 포화도를 측정할 수 있는 생체 센서 모듈을 사용하였다.

표 1. 시스템 모델
Table 1. System Model.

Software Platform	
Dev. Tool	Eclipse Java Juno Service Release 1
언어	JAVA JDK 1.7.0_25
Plug in	Android ADT
SDK	Android 4.2 (Jelly Bean)
Emulator	AVD16-wvga.avd
Hardware Platform	
PC	3세대 Intel Core i5 Pr. Notebook
OS	Windows 7 - Home Premium Edition
Web Container	Tomcat 7.0
Database	MySQL 5.5
Sensor Module	
Mote	ZigbeX II Mote,
Comm. Module	Bluetooth Module
Sensor Module	Motion 3X Module, TN9 Module

IV. 센서기반 건강 다이어트 응용 설계

이 장에서는 안드로이드 기반 스마트 기기를 위한 센서기반 건강 다이어트 응용의 주요 알고리즘을 제안한다. 본 논문에서 설계한 센서기반 건강 다이어트 응용은 사용자가 보다 효과적인 다이어트 수행을 할 수 있도록 안드로이드 OS 기반 스마트 폰을 이용하여 매일 섭취하는 음식의 종류와 양을 입력받아 총 섭취한 칼로리를 계산하고, 이렇게 섭취한 칼로리를 효과적으로 소모하기 위한 효율적인 운동 선택과 처방 및 소모 칼로리 관리를 수행하게 된다.

그림 1은 본 논문에서 제안한 센서기반 다이어트 어플에서의 사용자가 섭취한 음식과 다이어트를 위해 수행한 운동에 대한 입력 알고리즘을 보여준다. 음식과 운동에 대한 입력은 스마트 폰의 카메라 모듈을 이용하여 QR 코드의 형태로 정의된 음식과 수행한 운동에 대한 정보를 입력받아 그 종류에 따라 대분류, 중분류, 소분류로 구분하여 데이터베이스에 저장하고, 사용자가 섭취한 음식의 량과 수행한 운동의 강도에 따라 그 칼로리를 계산하여

```
public class Food extends Activity implements View.OnClickListener,
    OnCheckedChangeListener, OnItemSelectedListener {
    int type = 1; // 운동 음식 구분용
    String date;
    ArrayList<data> mydata = new ArrayList();
    // 내 데이터에 넣기 전에 임시 저장용
    ArrayAdapter adapter; // 소분류 내용 채우기 위한 어댑터
    ArrayAdapter<String> listadapter;
    // 리스트뷰에 채우기 위한 어댑터
    Spinner Large; // 대분류 스피너
    Spinner small; // 소분류 스피너
    ArrayList<String> list; // 소분류 채우기 위한 리스트
    ArrayList<String> items; // 리스트뷰 채우기 위한 리스트
    private static DataBase db; // 디비 객체
    int count; // 개수에 따라 저장하기 위한
    Button add; // 추가버튼
    Button modify; // 수정버튼
    Button remove; // 삭제 버튼
    Button save; // 저장버튼
    Button qrbtn; //바코드 버튼
    Button gpsbtn; //구글맵 버튼
    Button sensorbtn;
    EditText num; // 개수 에디트텍스트
    ListView lv; // 리스트뷰
    static LayoutInflater inflater;
    static FoodHealth foodHealth;
    static Context ctx;
    Intent intent;
    String result;
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.food);
        ctx = this;
        intent = getIntent();
        date = intent.getStringExtra("date");
        // 인텐트로 날짜 받아오기
        result = intent.getStringExtra("result");
        -중략-
        db = new DataBase(this); // 객체 생성
        db.open(); // 오픈
        rg.setOnCheckedChangeListener(this); // 리스너 사용
        Large = (Spinner) findViewById(R.id.spinner1);
        // 대분류 아이디 받기
        Large.setOnItemSelectedListener(this); // 리스너 사용
        small = (Spinner) findViewById(R.id.spinner2);
        // 소분류아이디 받기
        android.R.layout.simple_spinner_item);
        // 처음에 음식 대분류 채우기 위한
        -중략-
        save = (Button) findViewById(R.id.save);
        save.setOnClickListener(this);

        lv = (ListView) findViewById(R.id.listView1);
        // 리스트뷰 아이디 받기
        items = new ArrayList<String>(); // 리스트 생성
        listadapter = new ArrayAdapter<String>(this,
            R.layout.simpleitem, items);
        lv.setAdapter(listadapter);
        // 리스트뷰 사용하기 위해 어댑터 연결
        lv.setChoiceMode(lv.CHOICE_MODE_SINGLE);
        // 하나만 클릭하게 만들
        lv.setOnItemClickListener(mItemClickListener);
        // 리스너 사용
        inflater = (LayoutInflater)
            getSystemService(LAYOUT_INFLATER_SERVICE);
        try {
            Cursor c = db.readMYDB(date);
            listviewinit(c);
        } catch (Exception e) {
        }
    }
    -중략-
```

그림 1. 음식과 운동 입력 알고리즘
Fig. 1. Input Algorithm for Foods and Practices.

현재까지의 사용자의 음식 섭취 상태와 운동으로 소모된 칼로리의 상태를 보여주고, 추가로 효과적인 다이어트를 위한 음식 섭취나 운동 처방에 대한 의사결정 결과를 보

```

public class DataBase {
    private DBHelper mDbHelper;
    private SQLiteDatabase mDb;
    private final Context mContext;
    public DataBase(Context ctx) {
        this.mContext = ctx;
    }
    private static class DBHelper extends SQLiteOpenHelper {
        //안드로이드에서 sqlite 사용하기 위한 클래스
        public DBHelper(Context context) {
            super(context, "dietdiaryDB", null, 1);
        }
        public void onCreate(SQLiteDatabase db) {
            // 데이터베이스 생성부분
        }
        public void onUpgrade(SQLiteDatabase db, int
            oldVersion, int newVersion) {
            // 업그레이드부분
            db.execSQL("drop table if exists food");
            db.execSQL("drop table if exists health");
            db.execSQL("drop table if exists mydb");
            db.execSQL("drop table if exists memo");
            onCreate(db);
        }
    }
    public DataBase open() throws SQLException {
        // db 객체 가져오는부분
        mDbHelper = new DBHelper(mContext);
        mDb = mDbHelper.getWritableDatabase();
        return this;
    }
    public void close() {
        mDb.close();
        mDbHelper.close();
    }
    public ArrayList<String> readSC(String large, String dbtable)
        throws Exception { // 소분류 불러오기
        ArrayList<String> list = new ArrayList<String>();
        Cursor c = mDb.rawQuery("select 대분류, 소분류 from
            "+dbtable+" where 대분류 = '"+large+"';", null);
        }
        c.close();
        return list;
    }
    public void writeMYDB(String date, int cal, String large,
        String small, //내 정보 쓰기
        int type) {
        mDb.execSQL("insert into MYDB(대분류,소분류,
            ,type,value,날짜) values('"+large+"','"+small+"',
            + type + "','" + cal + "','" + date + "')");
    }
    //중략-
    public void writefood(String large, String small, String cal,
        String dbname){
        ContentValues values = new ContentValues();
        values.put("대분류", large);
        values.put("소분류", small);
        values.put("칼로리", cal);
        int colNum = mDb.update(dbname, values, "소분류 ='"
            + small + "'", null);
    }
    public Cursor readFood(String dbtable, String small) throws
        Exception {
        // 음식 db에서 데이터 가져오기
        Cursor c = mDb.rawQuery("select 대분류, 소분류 , 칼
            로리 from "+dbtable+" where 소분류 = '"+ small + "'", null);
        }
        public Cursor readMYDB(String date) throws Exception {
        //내 정보 읽어오기
        Cursor c = mDb.rawQuery("select 대분류, 소분류, type,
            날짜, value from MYDB where 날짜 = '"+ date + "'", null);
        }
    }
}
    
```

그림 2. 스마트 폰의 SQLite DB 연동 알고리즘
 Fig. 2. SQLite DB Interaction Algorithm in a Smart Phone.

여주게 된다. 그림 2는 입력 알고리즘에서 획득한 음식과 운동 정보를 저장하기 위해 스마트 폰의 SQLite DB와 연동하는 알고리즘을 보여준다.

```

/* Start the ConnectedThread to begin managing a Bluetooth
connection */
public synchronized void connected(BluetoothSocket socket,
    BluetoothDevice device, final String socketType) {
    if (D) Log.d(TAG, "connected, Socket Type:" + socketType);
    // Cancel the thread that completed the connection
    if (mConnectThread != null) {mConnectThread.cancel();
        mConnectThread = null;}
    // Cancel any thread currently running a connection
    if (mConnectedThread != null) {mConnectedThread.cancel();
        mConnectedThread = null;}
    // Cancel the accept thread because we only want to connect
    to one device
    if (mSecureAcceptThread != null) {
        mSecureAcceptThread.cancel();
        mSecureAcceptThread = null;
    }
    if (mInsecureAcceptThread != null) {
        mInsecureAcceptThread.cancel();
        mInsecureAcceptThread = null;
    }
    // Start the thread to manage the connection and perform
    transmissions
    mConnectedThread = new ConnectedThread(socket,
        socketType);
    mConnectedThread.start();
    // Send the name of the connected device back to the UI
    Activity
    Message msg = mHandler.obtainMessage(BluetoothChat.
        MESSAGE_DEVICE_NAME);
    Bundle bundle = new Bundle();
    bundle.putString(BluetoothChat.DEVICE_NAME,
        device.getName());
    msg.setData(bundle);
    mHandler.sendMessage(msg);
    setState(STATE_CONNECTED);
}
/* Stop all threads */
public synchronized void stop() {
    if (D) Log.d(TAG, "stop");
    if (mConnectThread != null) {
        mConnectThread.cancel();
        mConnectThread = null;
    }
    if (mConnectedThread != null) {
        mConnectedThread.cancel();
        mConnectedThread = null;
    }
    if (mSecureAcceptThread != null) {
        mSecureAcceptThread.cancel();
        mSecureAcceptThread = null;
    }
}
//중략-
    
```

그림 3. 스마트 폰에서의 블루투스 기반 연결 알고리즘
 Fig. 3. Bluetooth based Connection Algorithm in a Smart Phone.

본 논문에서 제안하는 어플은 건강한 다이어트를 수행할 수 있도록 하기 위해 현재 다이어트를 수행하고 있는 사용자의 건강 상태를 실시간으로 측정하여 반영할 수 있도록 블루투스 통신 모듈을 장착한 센서 모듈로부터 센싱된 맥박, 체온, 산소 포화도 등의 생체 데이터를 수신하여 현재의 사용자의 건강 상태 정보로 활용하게 하였다. 그림 3은 블루투스 통신 모듈을 장착한 센서 모듈로부터 센싱된 생체 데이터를 스마트 폰의 블루투스 모듈을 통해 수신하기 위해 스마트 기기와 블루투스를 통해 연결하는 알고리즘의 일부를 보여 준다. 센서 모듈로부터 송신된 데이터는 스마트 기기의 블루투스 모듈을

통해 실시간으로 수신되고 액정화면을 통해 수신된 생체 데이터를 디스플레이하게 하였고 또한 수신된 생체 데이터를 스마트 기기의 데이터베이스에 저장하고 현재의 사용자의 건강 상태 정보로 반영하게 하였다.

V. 구현 결과 및 고찰

본 논문에서 설계하고 구현한 스마트 응용은 사용자가 건강하고 보다 효과적인 다이어트 수행을 할 수 있도록 스마트 폰을 이용하여 매일 섭취하는 음식의 종류와 양을 입력받아 총 섭취한 칼로리를 계산하고, 이렇게 섭취한 칼로리를 효과적으로 소모하기 위한 효율적인 운동 종류의 선택과 운동 처방을 통해 소모 칼로리 관리를 하게 된다. 사용자의 지나치게 무리한 다이어트를 피하고 신체 건강을 해치지 않고 체중을 효과적으로 줄일 수 있는 건강한 다이어트를 유도하기 위하여 현재 수행중인 다이어트 프로그램에 채운, 맥박, 산소 포화도 등 사용자의 현재 생체 정보를 실시간으로 측정하여 다이어트를 위한 운동 처방에 직접적으로 반영할 수 있도록 하였다. 이를 위해 추가로 사용자 주변의 생체 데이터 측정 센서들과 블루투스 모듈을 통해 무선으로 연동하여 실시간으로 사용자의 생체 데이터를 측정하고 반영하여 건강 다이어트 프로그램에 적용할 수 있도록 하였다. 획득된 사용자의 생체 데이터는 스마트 폰의 데이터베이스에 저장하고 분석하여 의사결정 과정을 거쳐 사용자의 건강 상태를 결정하고 처방하는 건강한 다이어트를 유도하는 헬스케어 스마트 응용이다. 이 앱은 구글의 안드로이드 OS를 기반으로 하는 스마트 기기에 내장된 블루투스 모듈을 통해 다중 센서로부터 획득된 실시간 데이터를 데이터베이스에 저장하고 이를 바탕으로 사용자의 건강 상태를 판단하고 의사결정을 하는 헬스케어 스마트 응용 보조 계측장비로 이용될 수 있다.

그림 4는 본 논문에서 제안하고 설계한 센서 기반 건강 다이어트 어플의 초기 환경 설정을 하는 화면으로 각 사용자의 기초 대사량을 계산하고 수행하는 다이어트의 일정을 등록하는 화면을 보여준다. 그림에 보인바와 같이 사용자의 성별과 나이 및 신장, 몸무게 등의 신체 조건을 입력받아 각 사용자의 일일 기초 대사량을 계산하여 다이어트 어플에 반영하게 된다. 그림 5는 건강 다이어트 어플을 실행시키고 사용자가 그날 섭취한 음식과 수행한 운동을 입력하는 입력모듈 실행화면으로 섭취한



그림 4. 초기 설정 화면
Fig. 4. Execution Results of the Initial Setting Module.



그림 5. 음식과 운동 입력 모듈 실행결과
Fig. 5. Execution Results of Foods & Practices Input Module.



그림 6. 운동 결과와 생체 데이터 측정 모듈 실행결과
Fig. 6. Execution Results of Practice Map & Bionic Data Measurement Module.

음식이나 다이어트를 위해 수행한 운동에 대해 각 분류에 따라 입력하게 되어 있고, 이때 입력 노력을 절감시키기 위해 음식과 운동에 대한 QR 코드를 표준화하여 이를 스마트 폰의 카메라 모듈을 통해 손쉽게 입력할 수 있도록 하였다. 그림 6은 각각 스마트 폰 상에서 구동된 어플에서 구글 맵을 이용하여 조깅이나 걷기 운동 등을 수행하였을 때 운동 거리와 시간 그리고 운동 체중을 GPS를 이용하여 구글 지도상에 표시하도록 하였고, 이때 주변

의 다중 센서로부터 실시간으로 측정되어 전송되고 있는 사용자의 맥박, 체온, 산소 포화도 등 생체 데이터를 블루투스 모듈을 통해 수신하여 액정화면을 통해 그 결과를 디스플레이하는 실행 화면을 보여 준다. 주변의 센서 모드를 인식하고 초기화하여 주변의 다중 센서로부터 전송되는 생체 데이터를 감지하고, 전송되고 있는 데이터를 수신하여 수집된 생체 데이터는 데이터베이스에 저장되고 의사결정 과정을 거쳐 사용자의 건강 상태를 결정하고 건강 다이어트를 수행하게 하는데 반영되게 된다.

VI. 결 론

스마트 단말 기술의 급속한 발전으로, 스마트 폰을 이용한 다양한 어플들이 개발되고 다른 분야의 기술들과 융합되어 최근 스마트 응용의 주류를 이루고 있다.

본 논문에서는 스마트 폰을 위한 센서 기반 건강 다이어트 어플을 설계하고 구현하였다. 제한한 센서 기반 건강 다이어트 어플은 사용자가 보다 효과적인 다이어트 수행을 할 수 있도록 스마트 폰을 이용하여 매일 섭취하는 음식의 종류와 양을 입력받아 총 섭취한 칼로리를 계산하고, 이렇게 섭취한 칼로리를 효과적으로 소모하기 위한 효율적인 운동 처방 및 소모 칼로리 관리를 수행하게 하였다. 여기서 사용자의 건강을 해치지 않는 효과적인 다이어트를 유도하기 위하여 현재 수행중인 다이어트 프로그램에 체온, 맥박, 산소 포화도 등의 사용자 현재 건강 생체 정보를 실시간으로 측정하여 반영할 수 있도록, 추가로 주변의 생체 데이터 측정 센서들과 블루투스 기반의 무선 애드 혹 네트워크를 구축하고 무선으로 스마트 폰과 연동하여 다이어트 프로그램의 건강 다이어트 운동 처방을 위한 의사결정에 적용할 수 있도록 하였다. 향후 연구 과제로는 블루투스 모듈을 이용하여 수집된 생체 데이터를 이용하여 사용자의 현재 건강 상태를 정확히 판단하고 결정을 할 수 있는 의사결정 알고리즘에 관한 것이다.

References

[1] D. K. Lee, J. H. Kwon, "Social Search Algorithm considering Recent Interests of User", Journal of

Korean Institute of Information Technology, vol. 9, issue 4, pp. 187-194, 2011.

- [2] John, A., Adamic, L., Davis, M., Nack, F., Shamma, D. A., and Seligmann, D. D, "The future of online social interactions: what to expect in 2020", Proceedings of the 17th International Conference on WWW, 2008.
- [3] J. S. Seo, S. C. Park, "Design and Implementation of Support System for Personalized Medical Service Based on Mobile", The Journal of The Institute of Internet, Broadcasting and Communication, VOL. 13 No. 6, pp. 37-45, 2013.
- [4] S. Oh, "Design of a Smart Application Using Bluetooth based Ad-Hoc Sensor Networks", The Journal of the Institute of Webcasting, Internet and Telecommunication, Vol. 13, No. 6, pp. 243- 248, 2013.
- [5] S. Oh, "Design of a Middleware for Android-based Smartphone Applications", Journal of the Institute of Webcasting, Internet and Telecommunication, Vol., 12, No. 2, pp. 111 - 118, 2012.
- [6] http://inside.chosun.com/site/data/html_dir/2014/04/15/2014041500805.html
- [7] Reto Meier, Professional Android Application Development, John Wiley & Sons, Inc., pp. 864, 2012.
- [8] Greg Millette, Adam Stroud, Professional Android Sensor Programming, John Wiley & Sons, Inc., pp. 552, 2012.
- [9] <http://www.androidpub.com/1305>

저자 소개

오 선 진(중신회원)



- 제 6권 제2호 참조
- 현재 세명대학교 정보통신학부 교수
<주관심분야 : 스마트 응용, 그린IT, MANETs, 모바일컴퓨팅, USN 등>

※ 이 논문은 2013학년도 세명대학교 교내학술연구비 지원에 의해 수행된 연구임