



출품 분야 : 임베디드 SW, 윈도우, 모바일SW
기타()

※출품분야는 해당하는 분야에 밑줄을 치세요. 기타는 기입 후 밑줄

1. 프로젝트 이름

Body Shape

2. 팀 이름

Body Shape

3. 팀 구성

- 팀장 - 김장민(컴퓨터학부)
- 팀원 - 김동현(컴퓨터학부)
- 김진섭(컴퓨터학부)
- 정해원(컴퓨터학부)
- 최진호(컴퓨터학부)

4. 담당교수님

Yap Chee Een 교수님

5. 기획의도

1) 치열한 살과의 전쟁

지금 대한민국은 살과의 전쟁을 벌이고 있다. 어찌 보면 쓸쓸한 현실이지만 대한민국은 외모지상주의가 팽배하기 때문에 이러한 살과의 전쟁이 이뤄지고 있는 것이다. 다이어트 시장 규모는 무려 연 7조원에 육박하기 까지 하니 이 전쟁이 얼마나 치열한지는 굳이 설명하지 않아도 될 것이다. 하지만 대한민국 국민들은 살과의 전쟁을 벌이기엔 너무도 많은 방해물에 직면해 있기 때문에 이 전쟁은 주로 살의 승리로 끝나는 경우가 많다. 그 방해물에는 스트레스, 야식 등이 있지만 가장 큰 방해물은 뭘니 뭘니 해도 바로 부족한 시간과 비용에 있다. 우리는 다이어트를 하고자 하는 사람들과 부족한 시간과 비용 탓에 늘 작심삼일로 다이어트에 실패하는 사람까지도 모두 사로잡을 수 있는 것을 개발하고자 하였고 가장 손쉽게 접할 수 있는 어플리케이션을 개발하였다.

2) 시도에 비해 저조한 성공률

우리의 개발목적을 한 문장으로 정리하자면 성공적인 다이어트이다. 동기부여가 부족한 사람에게는 동기부여를 해주고 식단에 대한 지식이 없는 사람들을 위해서 다양한 다이어트 식단을 제공하고 운동을 하고 싶어 하는 사람들에게는 구체적인 운동법을 제공하는 등 다이어트가 작심삼일로 끝나지 않도록 해주는 어플리케이션을 개발하는 것이 목적이다.

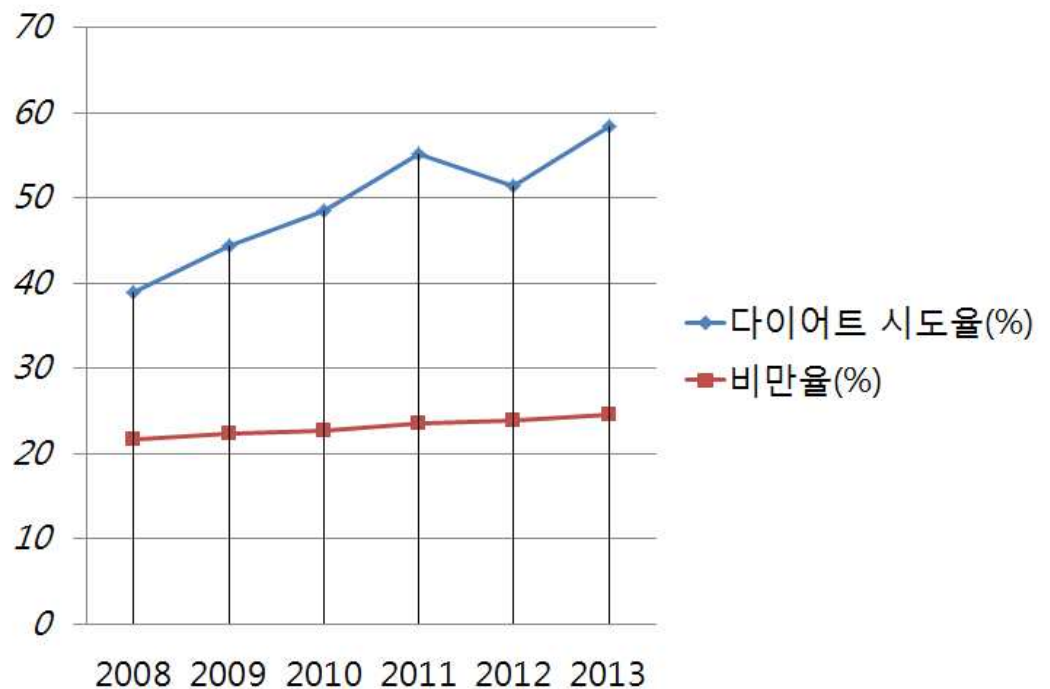


그림1) 연도별 다이어트 시도 및 비만 율

위의 표에서 볼 수 있듯 다이어트에 대한 시도 율은 급격하게 증가함에 반해, 비만 율의 변화는 크지 않음을 알 수 있다. 이에 따라, 다이어트의 성공이 성공적으로 이루어지지 않음을 알 수 있고, 우리는 좀 더 효율적이고 능동적인 다이어트 어플리케이션을 개발하려고 한다.

6. 프로젝트 및 작품 설명

가. 작품 개요

1) 심플한 디자인



그림 2) 삼성 S-Health (좌), 노 다이어트 트레이너 (우)

위의 두 어플리케이션 외에도, 다이어트 보조 어플리케이션의 경우 연령층에 구애 받지 않도록 단순하고 깔끔한 디자인을 제공하였고 Body Shape도 이처럼 이용하기 쉽도록 심플한 디자인을 채택하였다.

2) 기존과 달라진 기능

(1) 사진으로 몸의 변화를 확인할 수 있는 기능

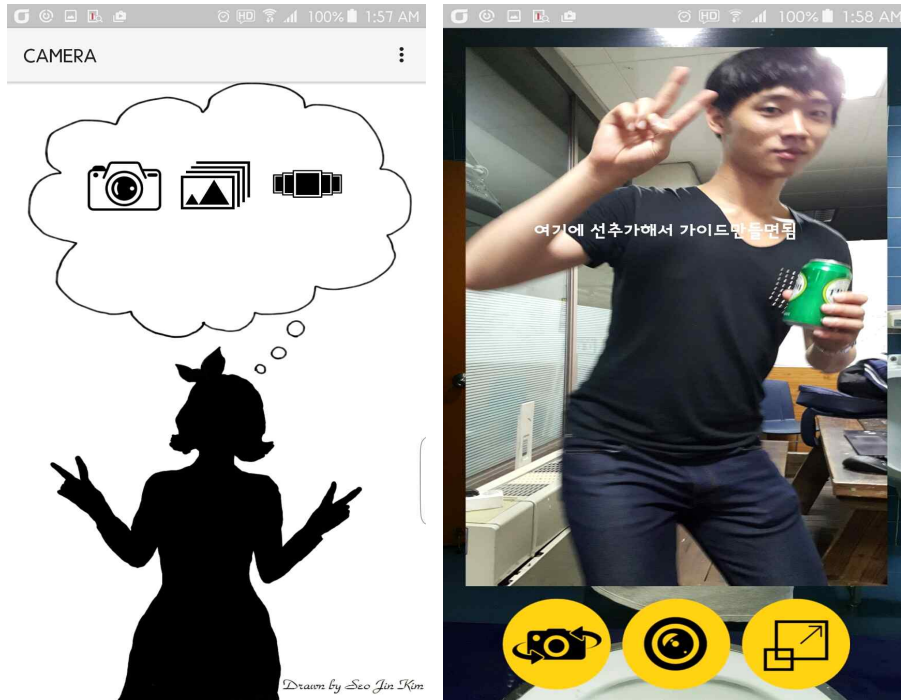


그림 3) Camera

그림 4) Camera-1. 촬영 및 바탕설정

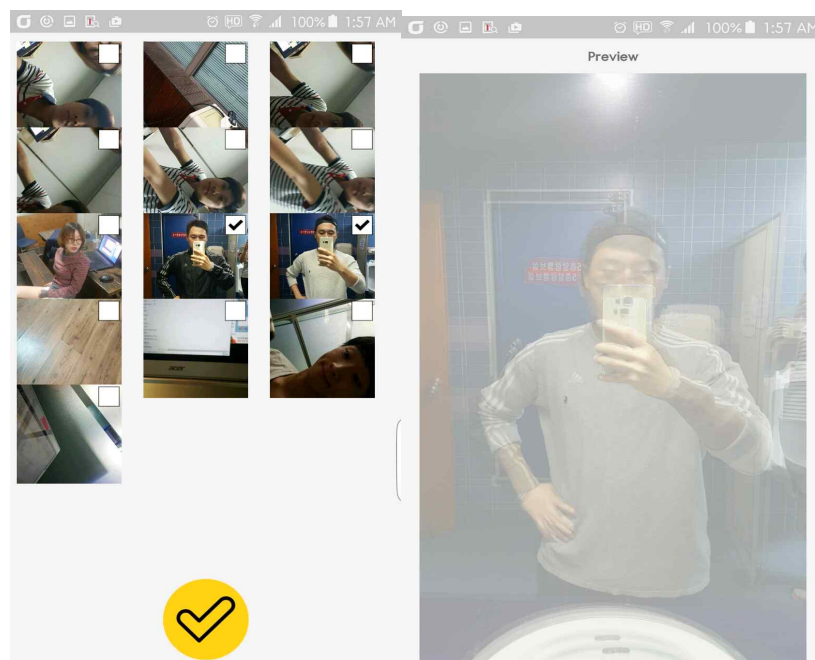


그림 5) Camera-2. Overlap 화면

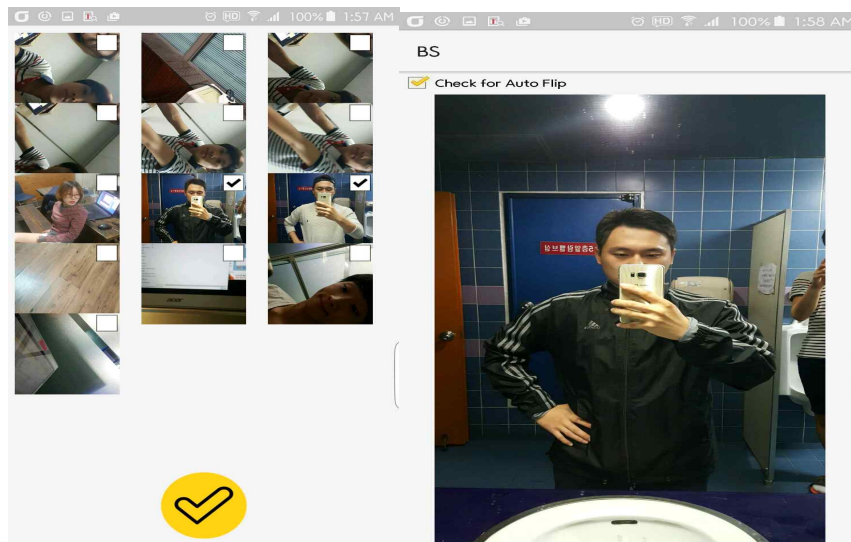


그림 6) Camera-3. Slide 화면

첫째, 카메라 화면에 가이드라인을 제공하여 항상 같은 자세로 같은 구도로 사진을 찍을 수 있도록 한다.

둘째, 사용자가 찍은 사진을 불투명하게 투명도를 조절하여 하나로 합치면 사용자의 몸이 얼마나 변화한지 알 수 있기 때문에 사진을 하나로 합치는 기능을 구현하고

셋째, 슬라이드 쇼를 구현하여 몸의 변화를 시각적으로 보여줄 것이다.

(2) 운동법 추천 기능

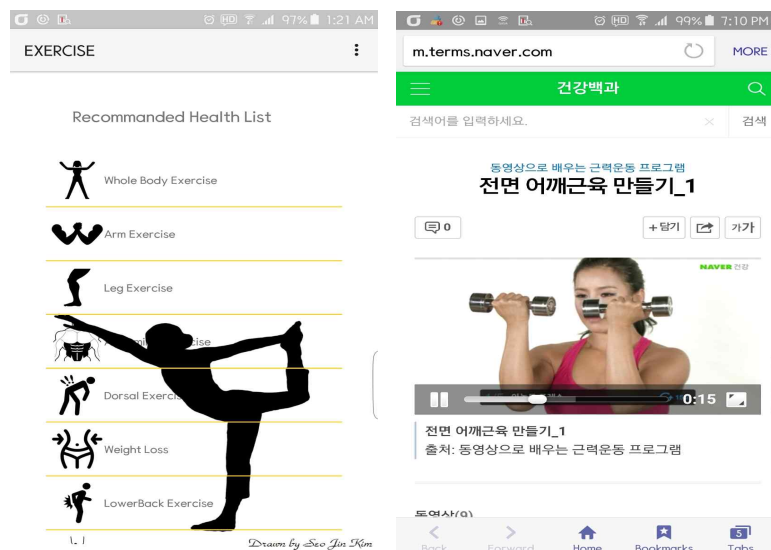


그림 7) Exercise-1. 운동 목록 리스트뷰 사진

그림 8) Exercise-2. 동영상 재생 Browser 사진

살을 빼고 싶은 부위나 근육을 강화하고 싶은 부분에 따라 다양한 운동법을 제공하고, 보다 구체적인 부위의 운동법을 사용자에게 제공한다.

(3) 식단 추천 기능

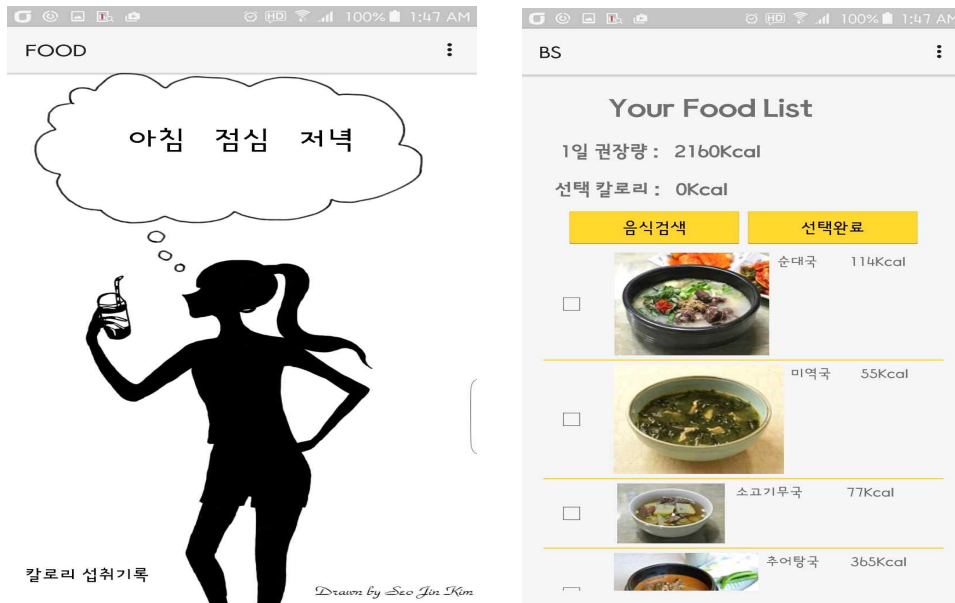


그림 9) FOOD-1. 시작 화면 그림 10) FOOD-2. 식단 입력

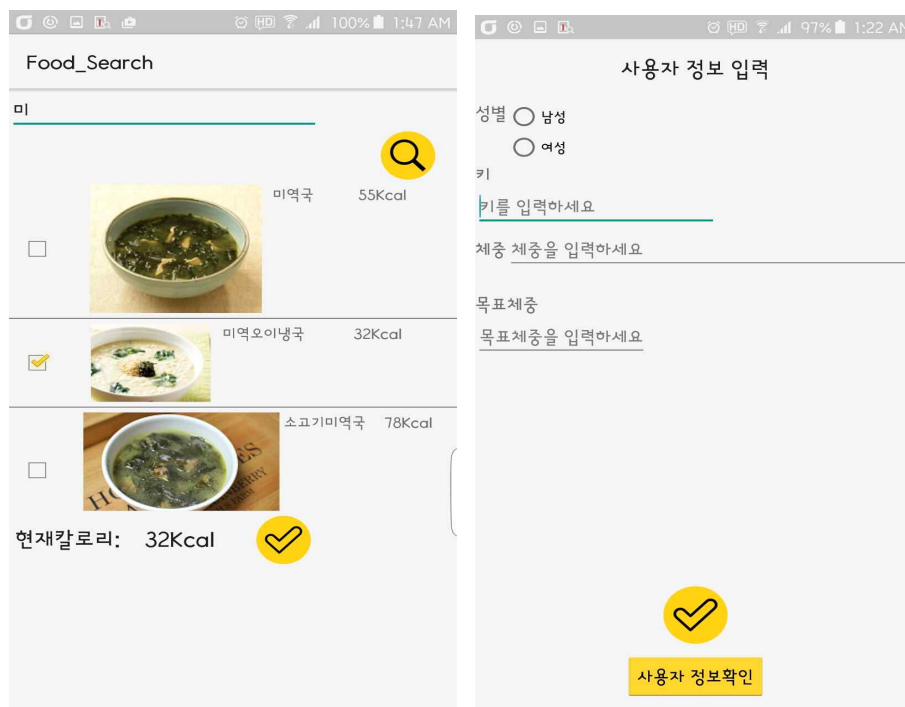


그림 11) FOOD-3. 음식 검색 화면
그림 12) MY STATUS-1. 사용자 정보 입력

다이어트에 가장 중요한 비중을 차지하는 식단을 추천하고 기록 할 수 있는 기능으로,

첫째, 사용자가 입력한 정보로 알맞게 계산 된 1일 권장 칼로리를 제공한다.

둘째, 동시에 섭취하는 식품을 리스트에서 능동적으로 검색, 선택할 수 있도록 하여 사용자가 식단을 수동 적으로 제공받을 경우 생기는 메뉴 선택에서의 제약을 없앴다.

셋째, 선택한 음식으로 아침, 점심, 저녁 식단을 기록 할 수 있다.

(4) 만보기 기능

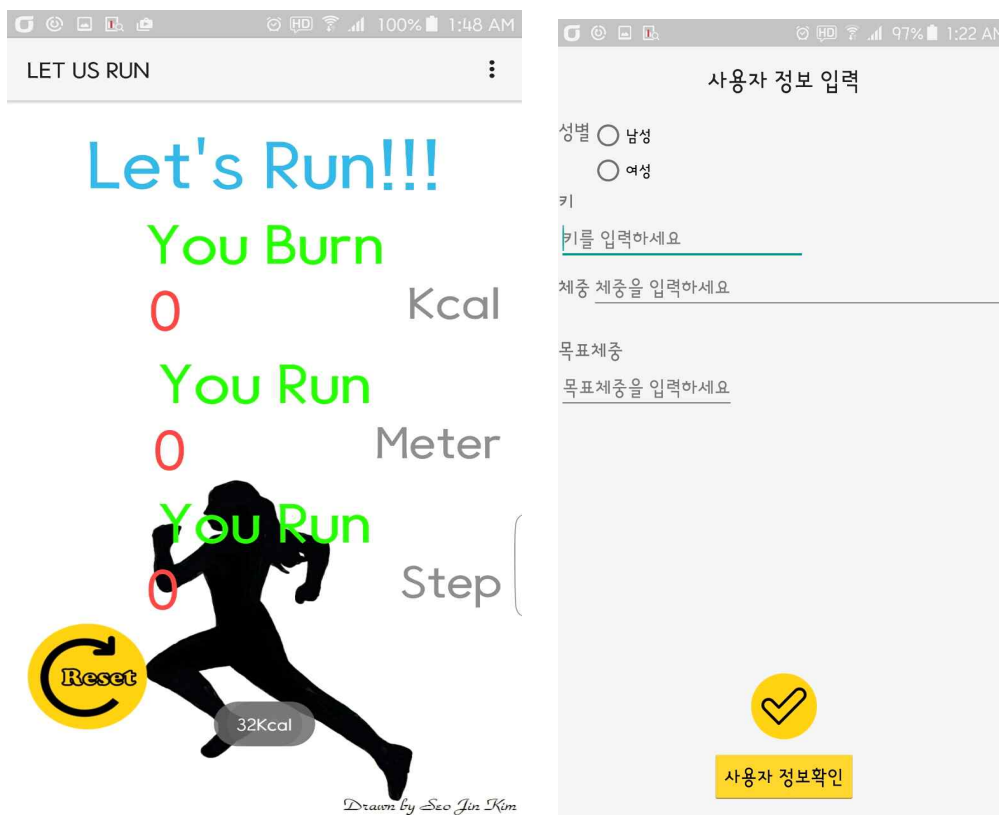


그림13) LET US RUN-1. 화면

그림 14) MY STATUS-1. 사용자 정보 입력

사용자의 원함에 따라, On/ Off 가 가능하도록 Stop Watch 방식으로 구현하였고, 사용자가 사전에 입력한 정보로 알맞게 계산

된 보폭에 따른 칼로리 소모량과 이동거리를 계산하여 사용자에게 제공한다.

	신체 수치	신체 변화 슬라이드 쇼	운동법 추천	식단 추천	만보기 기능	커뮤니티	유료/ 무료
다이어터	O	X	X	O	O	O	무료
다이어트 헌터	X	X	O	O	X	X	무료
홈다이어트	O	X	O	X	O	O	유료
Body Shape	O	O	O	O	O	X	무료

나. 시스템 구성도

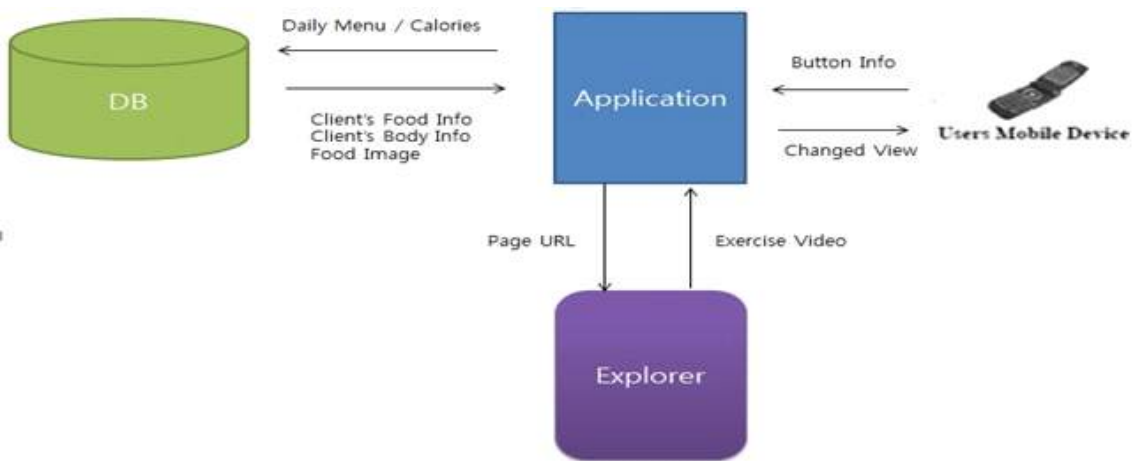


그림 15) System Architecture

다. 구현기능

(0) Main Page



그림 16) Navigation drawer UI 구성

어플리케이션 실행 시 가장 처음 사용자가 체험하는 Main Activity는 Navigation drawer UI가 적용되어 DrawerLayout내에 View를 선언하여 View들이 쌓이게 된다. 이 DrawerLayout은 크게 Navigation Drawer 뷰와 Main FrameLayout 뷰로 구성이 되어있다.

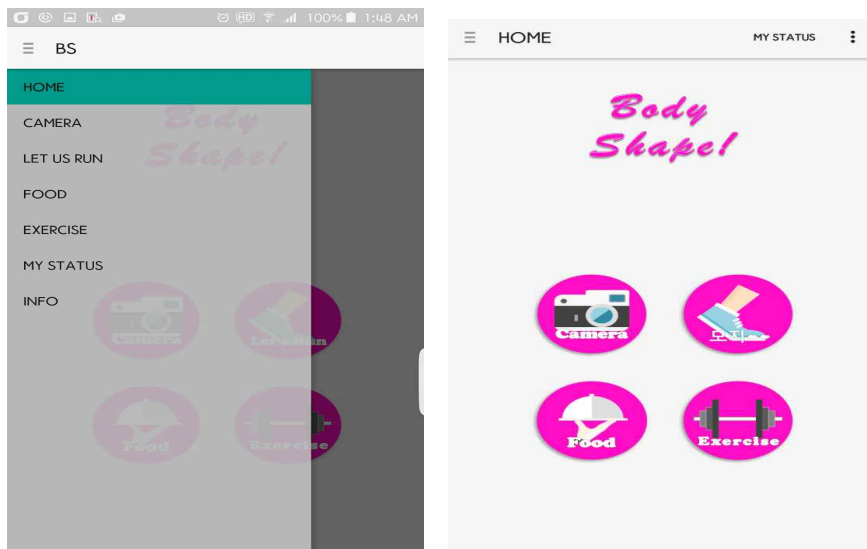


그림 17) Navigation Drawer의 본 어플리케이션의 적용 화면

그림 18) Main page

첫째, Navigation Drawer는 본 어플리케이션의 Activity들을 리스트로 나열하여 보여줌으로써, 사용자에게 더 직관적인 UI를 제공한다.

둘째, Main FrameLayout내에는 Camera, Manbo, Food, Exercises 버튼으로 구성되어있고, 각 버튼을 누르면 네 가지 주요 Activity의 시작 Activity를 실행한다. 상단 ActionBar는 사용자의 체형 세부사항을 입력받을 수 있는 MY STATUS와 INFO Activity를 실행한다.

(1) Camera

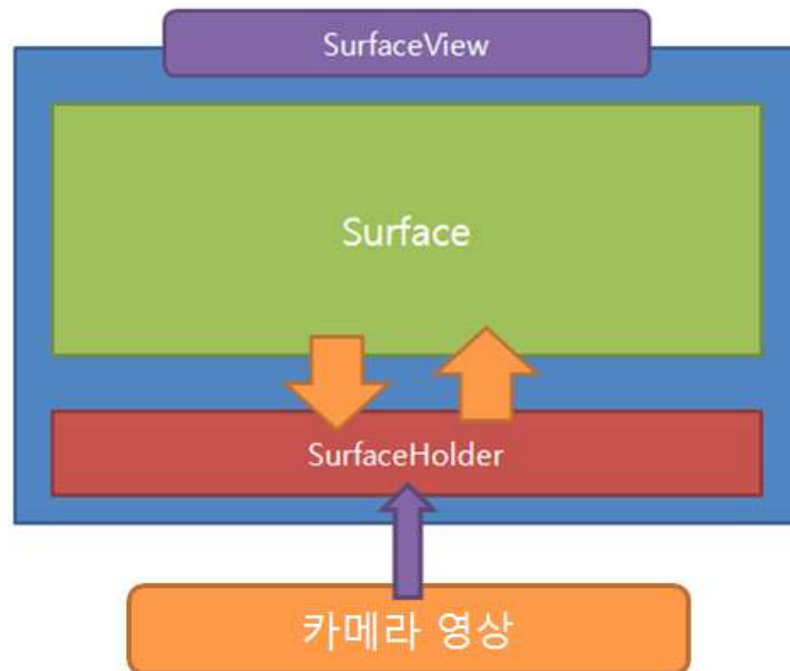


그림 19) SurfaceView

1. surfaceview를 이용한 카메라

우선 surfaceview는 imageview나 Textview처럼 콘텐츠를 표시할 수 있는 view들 중 하나다. 그러나 surfaceview를 이용하는 이유는 deviece에 장착되어있는 카메라를 통해 실시간으로 1초

에 몇십 프레임의 화면을 업데이트하기 위함인데 surfaceview가 아니라면 device의 thread 자원을 모두 써버려 어플리케이션이 정상적으로 작동을 안 할 수 있기 때문에 surfaceview를 사용한다. 이 그림에서 볼 수 있듯이 surfaceview는 하나의 틀의 역할을 해주고 그 안에 surface와 surfaceholder가 있다. 이중에 surfaceholder객체가 surface에 접근하여 화면을 처리하는데 surfaceholder가 카메라영상을 가져와 surface에서 화면을 처리하면 surface가 실제 우리가 보는 미리보기 화면을 보여준다.

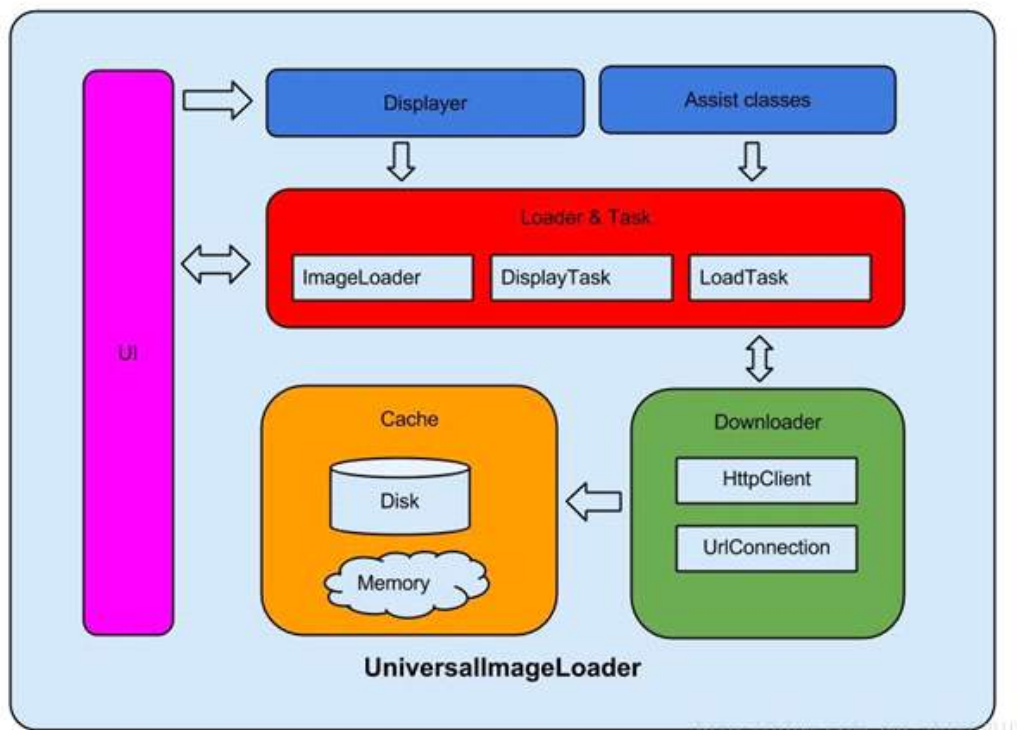


그림 20) ImageLoader

2. Imageloader

외부 라이브러리로 image를 좀더 쉽게 관리하고 가져올 수 있게 하는 라이브러리이다. 캐쉬디렉토리를 생성한 다음 디렉토리에 자신이 가져올 파일이 있는 경로를 넣어준다. 그 후 displayImageoption을 통해 캐쉬메모리의 사용여부와 사진을 불러올 타입과 bitmap 상태설정을 해준다.

3. overlap,slide

ImageLoader를 이용해 전체사진을 다 불러온 후 그 중에 내가 원하는 파일의 경로에 있는 파일들만 목록으로 불러오게 한다. 목록은 gridView와 arraylist를 이용해 불러오고, 10개 이하로 선택을 한다. Overlap의 경우 ImageView를 한 위치에 겹치게 만든 후 각각의 ImageView에 해당 경로에 있는 bitmap을 불러온다. Slide는 ViewFlipper와 Ontouch를 이용해 motion이 있을 경우 그에 해당하는 방향으로 sliding을 시켜 image를 바꾼다.

(2) 추천운동

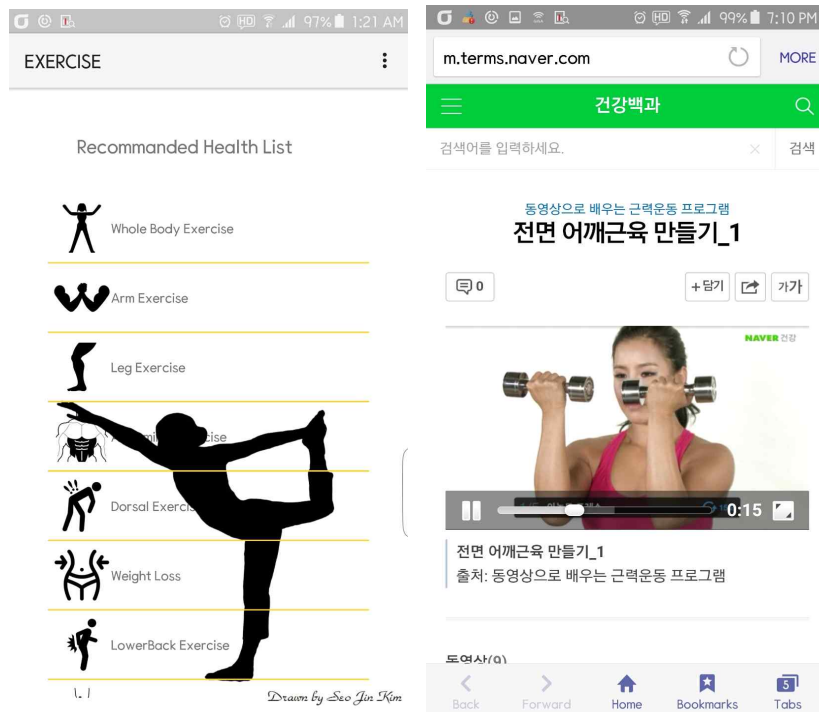


그림 21) Excercise-1. 운동 목록 리스트뷰 사진

그림 22) Excercise-2. 동영상 재생 Browser 사진

추천운동 기능은 안드로이드 스튜디오의 ListView 의 버튼 클릭 이벤트를 통하여 기존에 저장된 URL 링크를 브라우저를 통해

화면상에 띄울 수 있도록 하였다.

이 기능은 시중의 대부분의 어플리케이션에서 제공되는 기능이나, 대개 부위에 따른 운동이 정적으로 제공된다. 그래서 Body shape는 사용자가 같은 부위를 운동하더라도 본인에게 적합한 운동을 선택할 수 있도록, 최초 링크는 해당 부위의 기초 운동으로 보이되 사용자가 브라우저상에서 관련 운동들을 능동적으로 선택할 수 있도록 구현하였다.

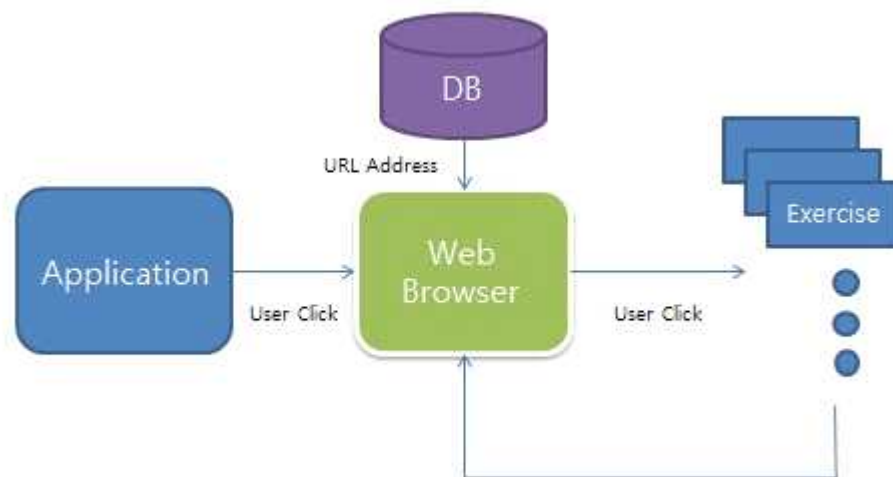
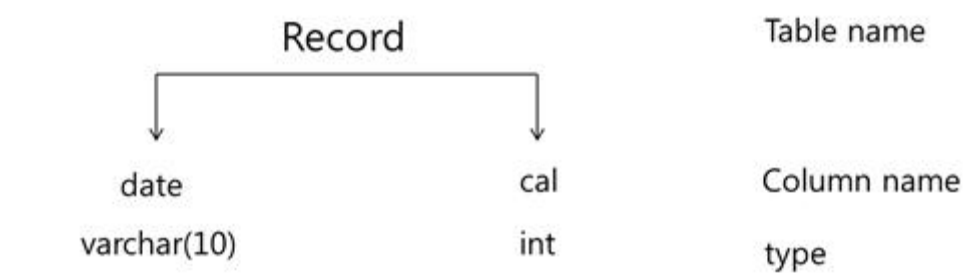


그림 23)Health System Architecture

3) 오늘의 음식

기능구현

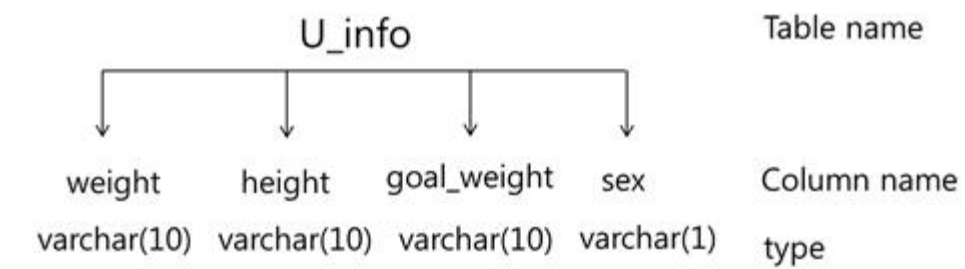
데이터베이스를 구현하기 위해 sqlite3를 이용하였다. 어플리케이션 특성상 한 번에 많은 프로세스가 접근 할 필요가 없고 원격 사용이 필요하지 않은 점 때문에 sqlite를 사용하였다. Body Shape 어플리케이션에 필요한 데이터베이스 테이블은 3개이다. 음식에 관한 정보를 담고 있는 food테이블, 그날 섭취한 칼로리를 저장한 record테이블, 사용자의 정보를 담은 u_info테이블이 있다.



date : “”으로 년 월 일을 나누어 날짜가 들어가있는 칼럼

Cal : 섭취한 칼로리량

1.테이블 구조

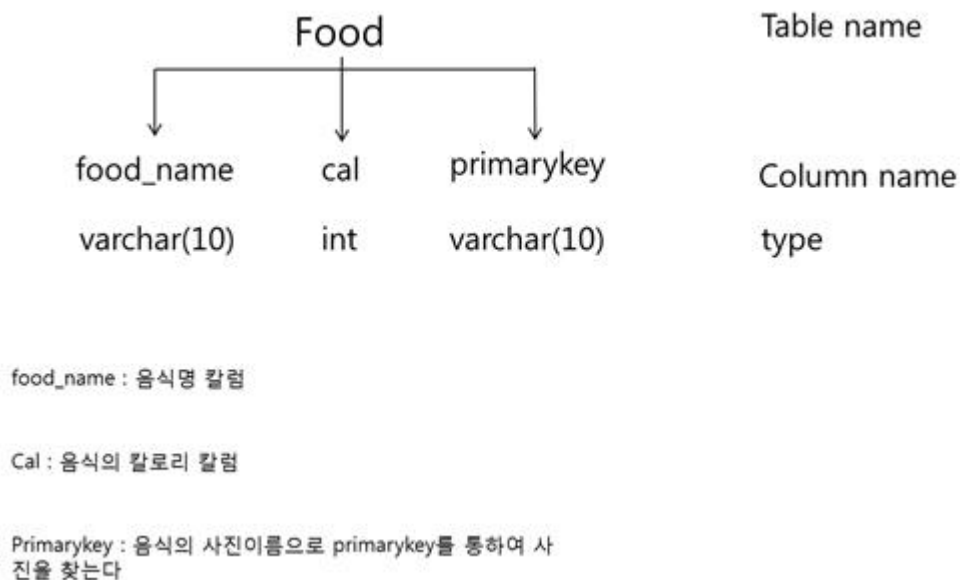


weight : 현재 몸무게 칼럼

height : 현재 키 칼럼

goal_weight : 목표 몸무게 칼럼

sex : 성별 칼럼(m/f로 구분)



2. 데이터베이스 적용

데이터베이스는 자기가 먹은 음식을 검색할 때, 만보계를 사용할 때 성별과 키 몸무게에 따라서 소모되는 칼로리가 다르게 계산 하기위해, 자신이 오늘 섭취한 칼로리의 양을 구할 때 사용한다.

SQLiteDatabase : SQLiteDatabase는 데이터베이스를 관리하는 메소드들을 제공한다.

1) **getWritableDatabase()** - 읽기 쓰기 모드로 데이터를 오픈한다.

2) **getReadableDatabase()** - 읽기전용 모드로 데이터베이스를 오픈한다.

Cursor

데이터베이스에서 코드로 데이터를 가져오는데 사용되는 것이 Cursor이다

Cursor는 커서가 가져온 데이터를 읽는 역할을 한다.

- 1) moveToFirst : 커서를 데이터의 맨 처음을 가리키게 한다.
- 2) moveToNext : 다음 데이터로 커서를 이동시킨다.

ContentValues

데이터베이스에 데이터를 입력할 때 사용한다.

- 1)put() : 파라미터로 데이터를 입력할 테이블과 값을 넣어준다.
- 2)insert() : 파라미터로 테이블명과 ContentValues의 객체가 들어간다.

SQLiteOpenHelper

생성자,데이터베이스를 생성하며 연결하는 object를 만들어준다.

(4) 만보기

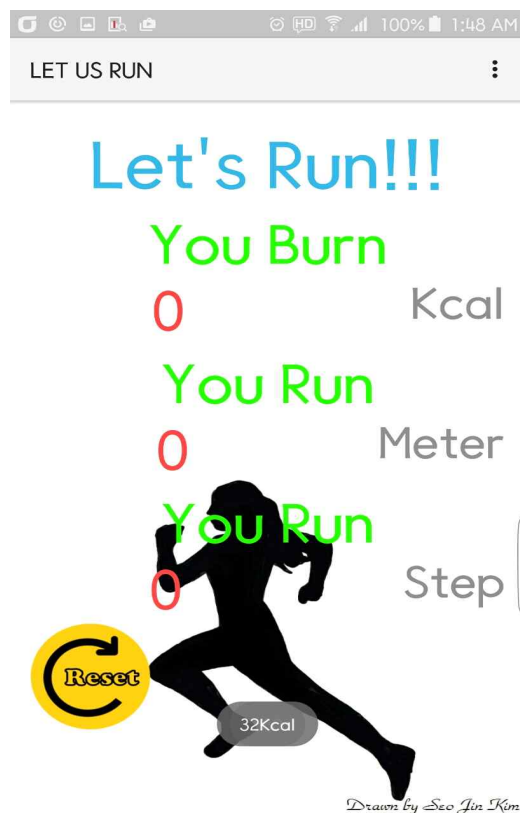


그림24) LET US RUN-1. 화면

달리기 기능을 구현하기 위해서 가속도 센서를 이용하였다. 가속도 센서를 이용하여 가속도가 변화함을 디바이스가 인지할 때마다 그것을 count 하여 숫자를 1씩 올리는 방식으로 구현하였다. 이 기능을 자세히 알아보기 위해 스마트폰의 가속도 센서에 대하여 가속도 센서에 대하여 알아보도록 하자. 우선 가속도 센서는 말 그대로 가속도를 측정하는 센서이다. 가속도 센서는 3차원에서 움직일 때 3축 즉 x , y , z 축을 이용하여 기울기가 달라지는 값을 측정해준다. 이때 기울기의 변화를 이용하여 어느 정도의 가속도가 생기는지 측정하게 되는데 스마트폰의 축은 아래 그림과 같이 형성되어있다.

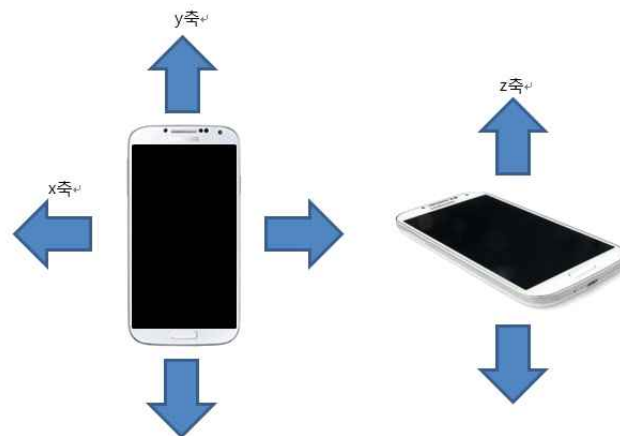


그림25) 가속도 센서 축 값

축에 대한 설명을 하자면 먼저 x축은 기기의 측면을 향한 움직임을 측정한다. 기기가 좌우로 수평 상태에서 움직임이 없으면 0 값을 가지며 왼쪽으로 기기가 움직이면 X축 값이 증가하고, 오른쪽으로 움직이면 감소한다. 가속도의 경우에는 실제 데이터와는 반대로 기기가 왼쪽으로 움직이면 가속도가 감소하고, 오른쪽으로 움직이면 증가하게 된다. 다음으로 y축은 기기의 전방/후방을 향한 움직임을 측정한다. 기기가 수평 한 상태에서 움직임이 없으면 0 값을 가지며 전방 즉, 기기의 윗부분으로 기기가 움직이면 Y축 값이 감소하

고, 후방으로 움직이면 Y축 값이 증가한다. 가속도의 경우에는 실제 데이터와는 반대로 기기가 전방으로 움직이면 가속도가 증가하고, 후방으로 움직이면 감소하게 된다. 마지막으로 z축은 기기의 위/아래를 향한 움직임을 측정한다. z축 데이터는 x축, y축과는 달리 중력의 영향을 받기 때문에 화면을 하늘로 향한 상태에서 수평을 유지하면 중력 값인 9.81 이 측정된다. 만약 z축의 순수 움직임을 측정하고 싶다면 z축 데이터에서 9.81 을 빼면 된다. 그러나 기기를 수직으로 배치하면 중력의 영향이 z축에서 y 축으로 이동하므로 기기의 배치에 맞는 축의 데이터에서 중력 값을 빼주어야 한다. 기기가 위쪽 즉, 하늘을 향해 움직이면 z축 값이 감소하고, 땅을 향해 이동하면 z축 값이 증가한다. 가속도의 경우에는 실제 데이터와는 반대로 기기가 위로 이동하면 가속도가 증가하고, 아래로 이동하면 감소하게 된다. 축과 축에 의한 센서 변화에 대해 알아보았으니 연산을 이용하여 가속도를 측정하는 방법에 대해 알아보겠다. 가속도 센서는 사물의 움직임을 측정하는데 이 움직임을 이용하여 다른 성질을 가지는 데이터로 가공할 수 있기 때문에 여러 가지 방법으로 연산을 하여 사용하곤 한다. 예를 들어 순수한 움직임만을 측정하는 경우에는 중력의 영향을 받고 있는 축의 데이터에서 9.81을 빼주어야 하고, 가속도를 측정하려면 2개 이상의 데이터를 모아서 연산을 해주어야 한다. 단순히 전달된 데이터를 출력할 때 데이터 값의 변화를 완충하려면 데이터에 저역 통과 필터(Low-Pass Filter : LPF)를 적용해야 한다. 저역 통과 필터를 사용할 때에는 결과 데이터를 추출하기 위해 사용되는 알파(alpha) 값이 필요한데 이것은 아래와 같은 연산결과로 산출된다. 이 알파 값을 구하는 공식은

$$\alpha = t / (t + dT)$$

다음과 같다.

t : 저속 통과 필터의 시정수. 시정수란 센서가 가속도의 63% 를 인지하는데 걸리는 시간을 말함.

dT : 이벤트 전송률 혹은 이벤트 전송속도.

위와 같은 alpha 값은 0.8 로 산출되며 alpha 에 직전 데이터를 곱하고, (1 - alpha) 에 현재 데이터를 곱하여 더해주면 필터링 된 움직임 값을 얻을 수 있다. 이렇게 계산하여 얻어진 가속도 값을 민감도를 설정하여 얼마나 가속도가 변하였을 때 카운트를 해줄 것 인지를 결정하여 구현하였다. 달리기를 할 때는 상당히 격한 움직임이 이루어지기 때문에 둔감하게 설정을 해두었다. 카운트 되는 만큼 달리기 걸음 수로 나타내었고 걸음 수 에 따른 칼로리와 거리를 이용자의 몸무게와 키를 고려한 연산 식을 이용하여 모두 나타내도록 구현하였다.

칼로리와 거리를 구하는 연산 식은 다음과 같다.

칼로리 = 몸무게 * 0.0009

거리 = 키 * 0.6

이렇게 달리기 측정 기능에서는 달린 걸음 수, 달린 거리, 달리는 동안 소모된 칼로리를 보여주어 사용자가 얼마만큼의 운동을 실시했는지 보여주게끔 하였다.

7. 개발환경

Android Studio 3.1.

SQLite 3

MS PowerPoint 2010