

Term Project 최종 보고서

<HealthMate>

컴퓨터 공학부
20133236
엄형근

프로젝트 목적

바쁜 생활로 인해 규칙적인 운동에 어려움이 있는 현대인들을 위해 하루의 운동 목표치를 하루 일과의 양에 따라 다르게 제공하고, 신체의 변화 모습을 기록하는 기능과 운동 방법에 대한 소개를 통해, 어플리케이션 하나만으로 운동 외의 불필요한 시간을 줄여줌으로서 운동 시간 확보에 도움을 준다.

구현 내용

자료구조 : 날씨 파싱 과정에서 노드 리스트 사용

센서 : Pedometer를 통한 accelerometer센서 사용

Polymorphism : Activity 상속과 OnClickListener 상속 과정에서의 수 많은 오버라이딩

DB, SQLite사용 : 요일별 목표량을 저장해놓는 DB 와 몸의 변화과정 확인에 필요한 리스트뷰에 사용된 DB

공공 DB 사용 : 기상청 api 사용 (RSS)

구현 기능별 UI 및 구현 과정

1. Splash Screen (로딩화면)

SKT 55% 오후 9:08



헬스 메이트

- 최초실행 시 3초간 라이언 캐릭터 anim.xml 실행후 메인 화면 출력
- Gif 파일을 바로 받는 데 어려움이 있어 이미지를 10장으로 나눠서 애니메이션 효과를 줍니다.

```
anim.xml x
1
<?xml version="1.0" encoding="utf-8"?>
<animation-list xmlns:android="http://schemas.android.com/apk/res/android"
    android:oneshot="false">

    <!-- oneshot = true면 한번만 false 면 무한반복 -->
    <item android:drawable="@drawable/healthmate_one"
        android:duration="75"/>
    <item android:drawable="@drawable/healthmate_two"
        android:duration="75"/>
    <item android:drawable="@drawable/healthmate_three"
        android:duration="75"/>
    <item android:drawable="@drawable/healthmate_four"
        android:duration="75"/>
    <item android:drawable="@drawable/healthmate_five"
        android:duration="75"/>
    <item android:drawable="@drawable/healthmate_six"
        android:duration="75"/>
    <item android:drawable="@drawable/healthmate_seven"
        android:duration="75"/>
    <item android:drawable="@drawable/healthmate_eight"
        android:duration="75"/>
    <item android:drawable="@drawable/healthmate_nine"
        android:duration="75"/>
    <item android:drawable="@drawable/healthmate_ten"
        android:duration="75"/>

</animation-list>
```

구현 기능별 UI 및 구현 과정

2. MainActivity (메인 화면)



- Navigation Bar와 오늘의 목표량 TextView, 각각의 액티비티로 들어갈 수 있는 이미지 버튼이 있습니다.
- <오늘의 목표량>으로 표시되어 있는 항목은 그날 그날 타임테이블 레이아웃에서 저장한 데이터베이스로부터 Calendar 객체를 통해 오늘의 요일에 따라 값을 받아옵니다.

```
// 무슨 요일인지에 따라 데이터베이스로부터 TimeTable.class에서 생성한 오늘의 목표량 받아오기
myDB = new MyDB(this);
sqliteDatabase = myDB.getReadableDatabase();
dayOfWeek = cal.get(Calendar.DAY_OF_WEEK);
if (dayOfWeek == 1) {
    sql_goal = "SELECT * FROM goalone";
} else if (dayOfWeek == 2) {
    sql_goal = "SELECT * FROM goaltwo";
} else if (dayOfWeek == 3) {
    sql_goal = "SELECT * FROM goalthree";
} else if (dayOfWeek == 4) {
    sql_goal = "SELECT * FROM goalfour";
} else if (dayOfWeek == 5) {
    sql_goal = "SELECT * FROM goalfive";
} else if (dayOfWeek == 6) {
    sql_goal = "SELECT * FROM goalsix";
} else if (dayOfWeek == 7) {
    sql_goal = "SELECT * FROM goalseven";
}

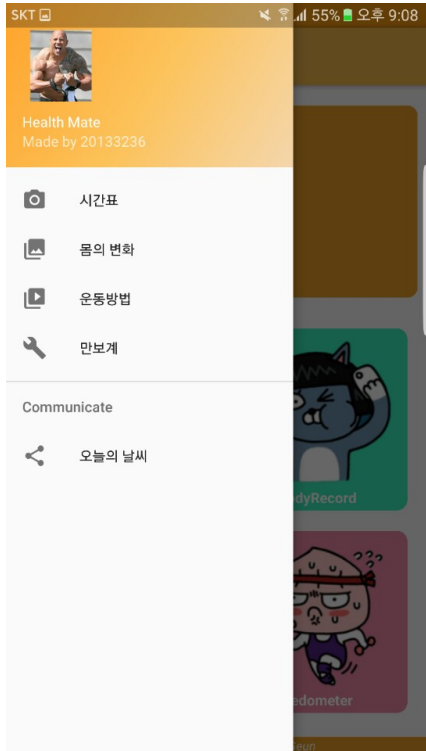
Cursor cursor = sqliteDatabase.rawQuery(sql_goal, null);
while (cursor.moveToNext()) {
    goal_str = "
    < 오늘의 목표량 > \n 팔굽혀펴기 : " + cursor.getString(1) + "\n 턱걸이 : " + cursor.getString(2) + "\n 윗몸일으키기 : " + cursor.getString(3) + "\n 어깨운동(분) : " + cursor.getString(4) + "\n 줄넘기 : " + cursor.getString(5) + "\n 걷기(보) : " + cursor.getString(6);
}

if(goal_str == null){
    today_exercise.setText("\nTimeTable에서 오늘의 운동 난이도를 선택해 주세요!\n");
}
else {
    today_exercise.setText(goal_str);
}
cursor.close();
sqliteDatabase.close();
// 목표량 받아오기 끝
```

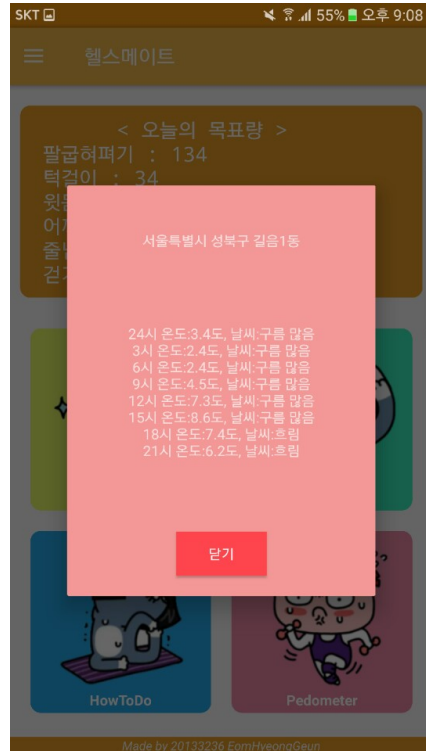
<(MainAcitivity.java) 무슨 요일인지에 따라 목표량 받아오는 코드>

구현 기능별 UI 및 구현 과정

3. Navigation Bar



- Navigation Bar -



-오늘의 날씨 클릭화면-

- 메인화면의 왼쪽 상단에 있는 버튼을 누르면 들어갈 수 있는 네비게이션 바를 구현하였습니다.
- Android Studio에서 제공하는 기본 Navigation Bar를 활용하였습니다.
- 시간표, 몸의 변화, 운동방법, 만보계를 클릭하면 각각의 Activity가 호출됩니다.
- 오늘의 날씨 : 공공DB 기상청(<http://www.kma.go.kr/>) api를 따온 부분으로서 현재 국민대학교가 있는 성북구의 날씨 정보 페이지를 파싱했습니다. 시간별 온도, 날씨만을 간단하게 보여지게 해놨습니다.

```
// Activity 실행과 동시에 웹서버에 접속하기 때문에 Thread를 사용해야 한다.
new Thread(){
    @Override
    public void run() {
        StringBuffer sBuffer = new StringBuffer();
        try{
            String urlAddr = "http://www.kma.go.kr/wid/queryDFSRSS.jsp?zone=1129066000";
            URL url = new URL(urlAddr);
            HttpURLConnection conn = (HttpURLConnection)url.openConnection();
            if(conn != null){
                conn.setConnectTimeout(20000);
                conn.setUseCaches(false);

                if(conn.getResponseCode() == HttpURLConnection.HTTP_OK) {
                    //서버에서 읽어오기 위한 스트림 객체
                    InputStreamReader isr = new InputStreamReader(conn.getInputStream());
                    //줄단위로 읽어오기 위해 BufferedReader로 감싼다.
                    BufferedReader br = new BufferedReader(isr);
                    //반복문 돌면서읽어오기
                    while (true) {
                        String line = br.readLine();
                        if (line == null) {
                            break;
                        }
                        sBuffer.append(line);
                    }
                    br.close();
                    conn.disconnect();
                }
            }
            // 결과값 출력해보기
            text_weather.setText(sBuffer.toString());
            xml = sBuffer.toString();
        }catch (Exception e){
            e.printStackTrace();
        }
        parse(); // parse 함수 아래에 구현
    }
}.start();
```

```
public void parse(){
    try{
        DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();
        DocumentBuilder documentBuilder = factory.newDocumentBuilder();

        //xml을 InputStream형태로 변환
        InputStream is = new ByteArrayInputStream(xml.getBytes());

        //document와 element는 w3c dom에 있는 것을 импорт한다.
        Document doc = documentBuilder.parse(is);
        Element element = doc.getDocumentElement();

        //읽어올 태그명 정하기
        NodeList items_hour = element.getElementsByTagName("hour");
        NodeList items_temp = element.getElementsByTagName("temp");
        NodeList items_wfKor = element.getElementsByTagName("wfKor");
        //자료형 누적시킬 StringBuffer객체
        StringBuffer sBuffer = new StringBuffer();

        //반복문을 돌면서 모든 데이터 읽어오기(8번만! => 3시간간격 8번 = 오늘 하루)
        for(int i=0; i<8; i++){
            //읽어온 자료에서 알고 싶은 문자열의 인덱스 번호를 전달한다.

            //시간 읽어오기
            Node item = items_hour.item(i);
            Node text = item.getFirstChild();

            //해당 노드에서 문자열 읽어오기
            itemValue = text.getNodeValue();
            sBuffer.append(itemValue+"시\ 온도:");

            //온도 읽어오기
            item = items_temp.item(i);
            text = item.getFirstChild();
            itemValue = text.getNodeValue();
            sBuffer.append(itemValue+"도, 날씨:");

            //날씨 읽어오기
            item = items_wfKor.item(i);
            text = item.getFirstChild();
            itemValue = text.getNodeValue();
            sBuffer.append(itemValue+"\n");
        }
    }
}
```

<(WeatherDialog.java) RSS 파싱 과정>

구현 기능별 UI 및 구현 과정

4. TimeTable (시간표)

	일	월	화	수	목	금	토
난이도	11	12	13	14	15	16	17
팔굽혀펴기	115	93	88	140	54	81	134
턱걸이	38	23	26	35	12	23	34
윗몸일으키기	174	124	139	184	74	145	160
어깨문동(분)	29	16	18	24	12	18	22
줄넘기	347	212	204	319	127	223	338
걷기(보)	8439	5581	5920	8210	2862	5884	8651

난이도선택

운동 난이도를 선택하세요.

상 중 하

- 설정버튼 클릭하면 -

- 달력은 제가 직접 날짜 계산하는 식으로 주차 별로 표시되게 따로 만들었습니다.

```
// 오늘의 날짜 기준으로 계산하여 만든 나만의 달력, - 일주일만 표시됩니다 -
public void getDate(){
    Calendar calendar = Calendar.getInstance();
    int day = cal.get(Calendar.DAY_OF_MONTH)+1;
    int month = cal.get(Calendar.MONTH)+1;
    int year = cal.get(Calendar.YEAR);
    today_date.setText(year + "년 " + month + "월 " + (day-1) + "일 ");

    int week = cal.get(Calendar.DAY_OF_WEEK);
    int count = -week; // 무슨요일이냐에 따라 달력 출력이 달라지게 하기 위해

    sunday.setText((day + count++) + "");
    monday.setText((day + count++) + "");
    tuesday.setText((day + count++) + "");
    wednesday.setText((day + count++) + "");
    thursday.setText((day + count++) + "");
    friday.setText((day + count++) + "");
    saturday.setText((day + count++) + "");
}
```

- 하루의 여유시간 정도에 따라 난이도 설정 가능합니다. (각각 난수를 주어 난이도가 주어지면 일정 범위 내의 숫자가 전달됩니다.)

```
// 운동 난이도에 따라 난수를 다르게 주어 각 운동의 횟수가 다르게 각 운동의 데이터베이스에 입력했습니다.
private AlertDialog.Builder createDialogBox(final int dayOfWeek){
    AlertDialog.Builder builder = new AlertDialog.Builder(this);

    builder.setTitle("난이도선택");
    builder.setMessage("운동 난이도를 선택하세요.");

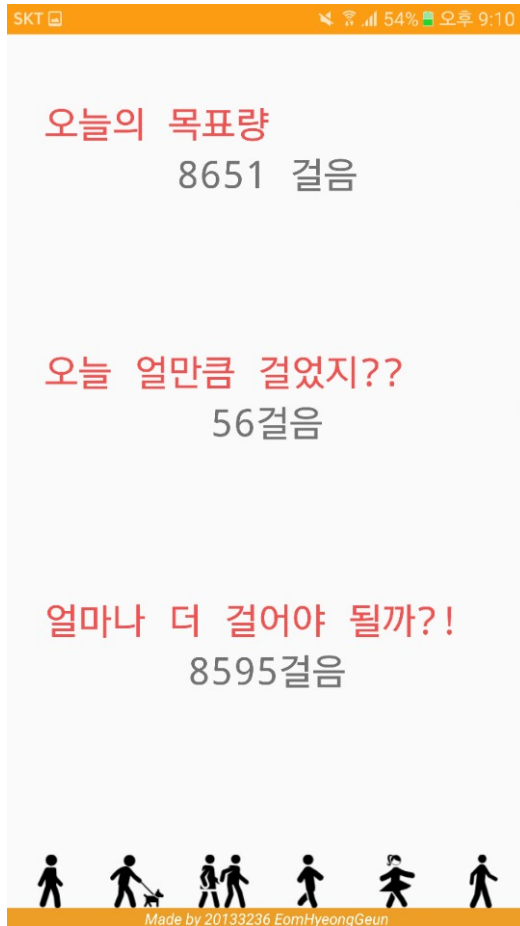
    builder.setNegativeButton("중", (dialog, which) -> {
        int pushup, chinup, situp, shoulder, rope, walk;
        pushup = (int) (Math.random()*30)+70; // 70~100개
        chinup = (int) (Math.random()*10)+20; // 20~30개
        situp = (int) (Math.random()*50)+100; // 100~150개
        shoulder = (int) (Math.random()*5)+15; // 15~20회
        rope = (int) (Math.random()*50)+200; // 200~ 250회
        walk = (int) (Math.random()*1000)+5000; // 5000~6000걸음

        sqLite = mydb.getWritableDatabase();
        if(dayOfWeek == 1) {
            sql_dialog = "INSERT INTO goalone(difficulty,pushup,chinup,situp,shoulder,rope,walk) " +
                "VALUES('2', " + pushup + ", " + chinup + ", " + situp + ", " + shoulder + ", " + rope + ", " + walk + ") ";
        }
        else if(dayOfWeek == 2) {
            sql_dialog = "INSERT INTO goaltwo(difficulty,pushup,chinup,situp,shoulder,rope,walk) " +
                "VALUES('2', " + pushup + ", " + chinup + ", " + situp + ", " + shoulder + ", " + rope + ", " + walk + ") ";
        }
        else if(dayOfWeek == 3) {
            sql_dialog = "INSERT INTO goalthree(difficulty,pushup,chinup,situp,shoulder,rope,walk) " +
                "VALUES('2', " + pushup + ", " + chinup + ", " + situp + ", " + shoulder + ", " + rope + ", " + walk + ") ";
        }
        else if(dayOfWeek == 4) {
            sql_dialog = "INSERT INTO goalfour(difficulty,pushup,chinup,situp,shoulder,rope,walk) " +
                "VALUES('2', " + pushup + ", " + chinup + ", " + situp + ", " + shoulder + ", " + rope + ", " + walk + ") ";
        }
        else if(dayOfWeek == 5) {
            sql_dialog = "INSERT INTO goalfive(difficulty,pushup,chinup,situp,shoulder,rope,walk) " +
                "VALUES('2', " + pushup + ", " + chinup + ", " + situp + ", " + shoulder + ", " + rope + ", " + walk + ") ";
        }
        else if(dayOfWeek == 6) {
            sql_dialog = "INSERT INTO goalsix(difficulty,pushup,chinup,situp,shoulder,rope,walk) " +
                "VALUES('2', " + pushup + ", " + chinup + ", " + situp + ", " + shoulder + ", " + rope + ", " + walk + ") ";
        }
        else if(dayOfWeek == 7) {
            sql_dialog = "INSERT INTO goalseven(difficulty,pushup,chinup,situp,shoulder,rope,walk) " +
                "VALUES('2', " + pushup + ", " + chinup + ", " + situp + ", " + shoulder + ", " + rope + ", " + walk + ") ";
        }
        sql_dialog = sql_dialog;
    });
}
```

<(TimeTableActivity.java) 운동 난이도에 따른 목표량 부여>

구현 기능별 UI 및 구현 과정

5. Pedometer (만보계)



- Accelerometer Sensor를 사용하여 만보계를 구현하였습니다.
- 오늘의 요일에 따라 Database에 있는 해당 요일의 table에서 목표 걷기의 양을 받아옵니다.
- onSensorChanged에서 event값(가속의 변화)에 따라 count를 증가시키고 그 count만큼 오늘의 목표량에서 줄어 들게 했습니다.

```
@Override
public void onSensorChanged(SensorEvent event) {
    if (event.sensor.getType() == Sensor.TYPE_ACCELEROMETER) {
        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;

            if (speed > SHAKE_THRESHOLD) {
                tv_count.setText("" + (++cnt) + "걸음");

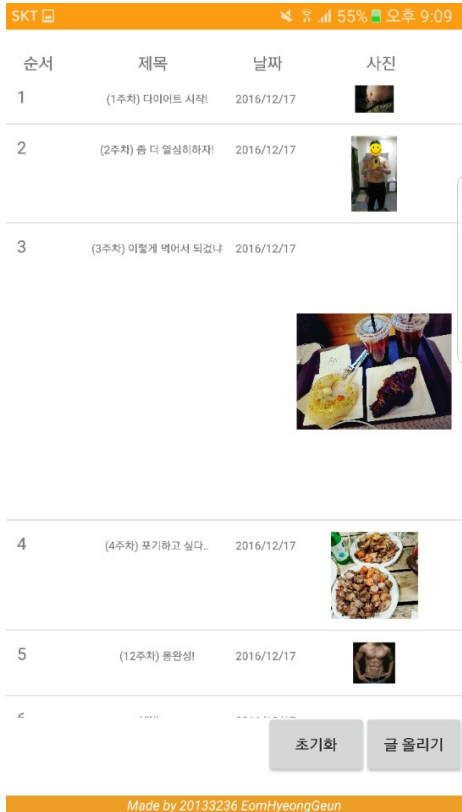
                // 목표량이 설정되지 않은 상태에서 pedometer 실행 시 목표량 설정해달라는 Toast와 함께 메인페이지로.
                try {
                    tv_remain.setText("" + (Integer.parseInt(goal_str) - cnt) + "걸음");
                } catch (Exception e) {
                    finish();
                    Toast.makeText(getApplicationContext(), "목표량을 설정해주세요!", Toast.LENGTH_SHORT).show();
                }
            }

            lastX = event.values[DATA_X];
            lastY = event.values[DATA_Y];
            lastZ = event.values[DATA_Z];
        }
    }
}
```

<(PedometerActivity.java)의 onSensorChanged>

구현 기능별 UI 및 구현 과정

6. BodyRecord (몸의 변화 기록 게시판)



- 제목/날짜/사진으로 구성된 리스트뷰
- 사진은 Uri를 String형태로 데이터베이스에 저장해두었다가 리스트뷰 아이템의 ImageView에서 이미지 장착할때 Uri파싱을 통해 보이게 했습니다.

```
// 리스트뷰 아이템들 변경
@Override
public View getView(int position, View convertView, ViewGroup parent) {
    final int pos = position;
    final Context context = parent.getContext();

    // "listview_item" Layout을 inflate하여 convertView 참조 획득.
    if (convertView == null) {
        LayoutInflater inflater = (LayoutInflater) context.getSystemService(Context.LAYOUT_INFLATER_SERVICE);
        convertView = inflater.inflate(R.layout.list_view_item_layout, parent, false);
    }

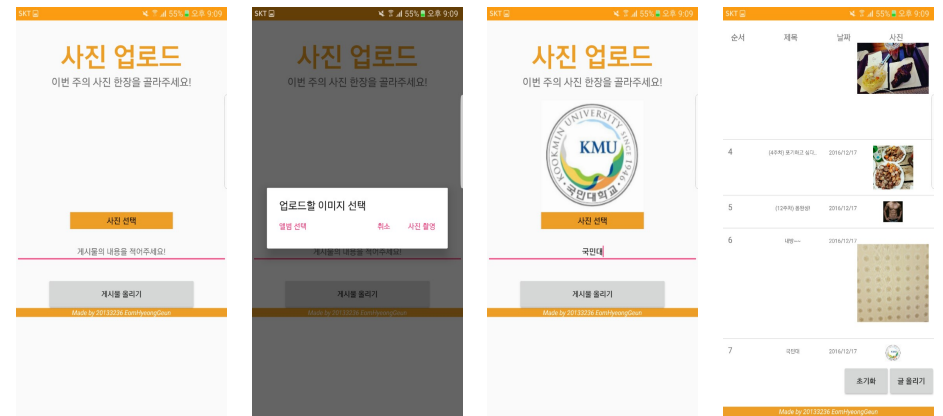
    ImageView pictureImg = (ImageView)convertView.findViewById(R.id.list_img);
    TextView idText = (TextView)convertView.findViewById(R.id.list_text1);
    TextView titleText = (TextView)convertView.findViewById(R.id.list_text2);
    TextView dateText = (TextView)convertView.findViewById(R.id.list_text3);

    ListViewItem listViewItem = listViewItemList.get(position);

    pictureImg.setImageURI(Uri.parse(listViewItem.getPicture()));
    idText.setText(listViewItem.getIdNum() + "");
    titleText.setText(listViewItem.getTitle());
    dateText.setText(listViewItem.getDate());

    return convertView;
}
```

<(ListViewAdapter.java)의 getView>



- 글 올리기 과정 -

```
@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    super.onActivityResult(requestCode, resultCode, data);

    if(resultCode != RESULT_OK)
        return;

    switch (requestCode){
        // 앨범선택 클릭시
        case PICK_FROM_ALBUM:
        {
            mImageCaptureUri = data.getData();
            Log.d("HealthMate", mImageCaptureUri.getPath().toString());
        }
        // 사진촬영 선택시
        case PICK_FROM_CAMERA:
        {
            // 일단 이미지를 가져오고 이후에 리사이징할 이미지의 크기를 결정
            Intent intent = new Intent("com.android.camera.action.CROP");
            intent.setDataAndType(mImageCaptureUri, "image/*");

            // CROP할 이미지를 200*200의 크기로 저장
            intent.putExtra("outputX", 200); // CROP할 이미지의 x축 크기
            intent.putExtra("outputY", 200); // CROP할 이미지의 y축 크기
            intent.putExtra("aspectX", 1); // CROP할 이미지의 x축 비율
            intent.putExtra("aspectY", 1); // CROP할 이미지의 y축 비율
            intent.putExtra("scale", true);
            intent.putExtra("return-data", true);

            startActivityForResult(intent, CROP_FROM_IMAGE); // CROP_FROM_IMAGE case문으로 이동
            break;
        }
        case CROP_FROM_IMAGE:
        {
            // 크롭이 된 이후의 이미지를 넘겨 받고 이미지 뷰에 이미지를 보여준다거나 부가적 작업이후 임시파일 삭제
            if(resultCode != RESULT_OK){
                return;
            }

            final Bundle extras = data.getExtras();

            //Crop된 이미지를 저장하기 위한 File 경로
            filePath = Environment.getExternalStorageDirectory().getAbsolutePath()
                    + "/HealthMate/" + System.currentTimeMillis() + ".jpg";

            if(extras != null){
                Bitmap photo = extras.getParcelable("data"); // CROP된 Bitmap
                iv_UserPhoto.setImageBitmap(photo); // 레이아웃의 이미지에 CROP된 BITMAP을 보여줌
                storeCropImage(photo, filePath); // CROP된 이미지를 외부저장소, 앨범에 저장한다
                absolutePath = filePath;
                break;
            }

            // 임시 파일 삭제
            File f = new File(mImageCaptureUri.getPath());
            if(f.exists()){
                f.delete();
            }
        }
    }
}
// onActivityResult 종료
```

<(UploadActivity.java)의 onActivityResult>

- 사진 선택 버튼을 누르면 앨범에서 사진을 가져오거나 사진 촬영을해서 업로드를 할 수 있는 Dialog가 호출됩니다.
- 사진을 촬영하는 경우에는 onAcitivityResult함수의 Override 과정에서 Intent 전달을 통해 Crop image(사진크기조절화면)으로 넘어갑니다.

구현 기능별 UI 및 구현 과정

7. HowToDo (운동방법)



- 각 부위 클릭 시 -

- HowToDo의 메인화면 사진에서 각 부위 클릭시 각 부위별 운동 방법과 운동 동영상을 제공하려고 했었습니다.
- 개발 시간의 부족으로 인하여 각각 해당하는 Activity만 만들어서 위치를 클릭하면 해당 Activity를 호출하게 했습니다. (TextView의 위치조정으로 맞췄습니다..)

아쉬웠던 점

- 새로그침 기능 구현을 하지 못했다. Activity LifeCycle을 이용하여 화면 전환시 그때그때 데이터베이스로부터 값을 가져오고 싶었지만, 너무 어려워서 좀 더 보완해야 할 것 같다.
- HowToDo 기능이 재밌고 기대되는 기능이었는데, 개발 시간 부족으로 인해 마무리 짓지 못하고 제출하는 것이 너무 아쉬워서, 종료 후에 추가적으로 개발을 해서 개발자가 이 앱을 유용하게 사용할 수 있게 할 것이다.
- Navigation Bar가 모든 Activity에서 뜨게 하는 것이 목적이었는데, 내가 생각하는 방향은 프로그램의 무게를 너무 무겁게 하고, 오픈소스들을 보니 너무 복잡해서 포기하게 되었다.
- 카카오톡 캐릭터 무단 도용으로 다른 사람들과 공유는 할 수 없을 것 같다..

Term Project GitHub Address

- <https://github.com/EomHyeongGeun/MobileProgrammingProject>