Yam.

학생식당 식단표 알림 및 커뮤니케이션 어플리케이션

소스 리뷰 및 전반적인 시스템 설명

포트폴리오

이우경

목차 a table of contents

YAM

Part 1 어플리케이션 설명

Part 1

어플리케이션 설명 - YAM

YAM?

경상대학교 학생식당 식단 알림 및 커뮤니티 어플리케이션

Part 1 어플리케이션 설명 - 주요기능

- 학생들의 NEEDS

1. 특정시간 학생 식당을 이용하지 않는 학생들을 위한 식사 시간 별 선택 알림 기능
2. 식단의 선호도를 볼 수 있도록 해주는 좋아요 기능 및 후기 커뮤니티 기능
3. 때때로 변하는 공지를 어플리케이션에 공지해주는 기능

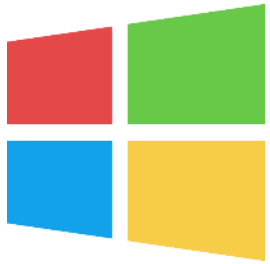
- 필요한 기술 및 라이브러리

1. Jsoup - 웹 크롤링 및 파싱 라이브러리
2. Json Simple - http 통신 시 json 형식으로 가공해주는 라이브러리
3. AWS Notification 기능

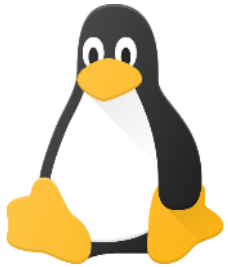
Part 2 개발 환경

Part 2 개발환경

OS환경



WINDOWS



LINUX SERVER

TOOL

ECLIPSE

ANDROID STUDIO

SQL DEVELOPER

그 외 환경 및 언어

JAVA

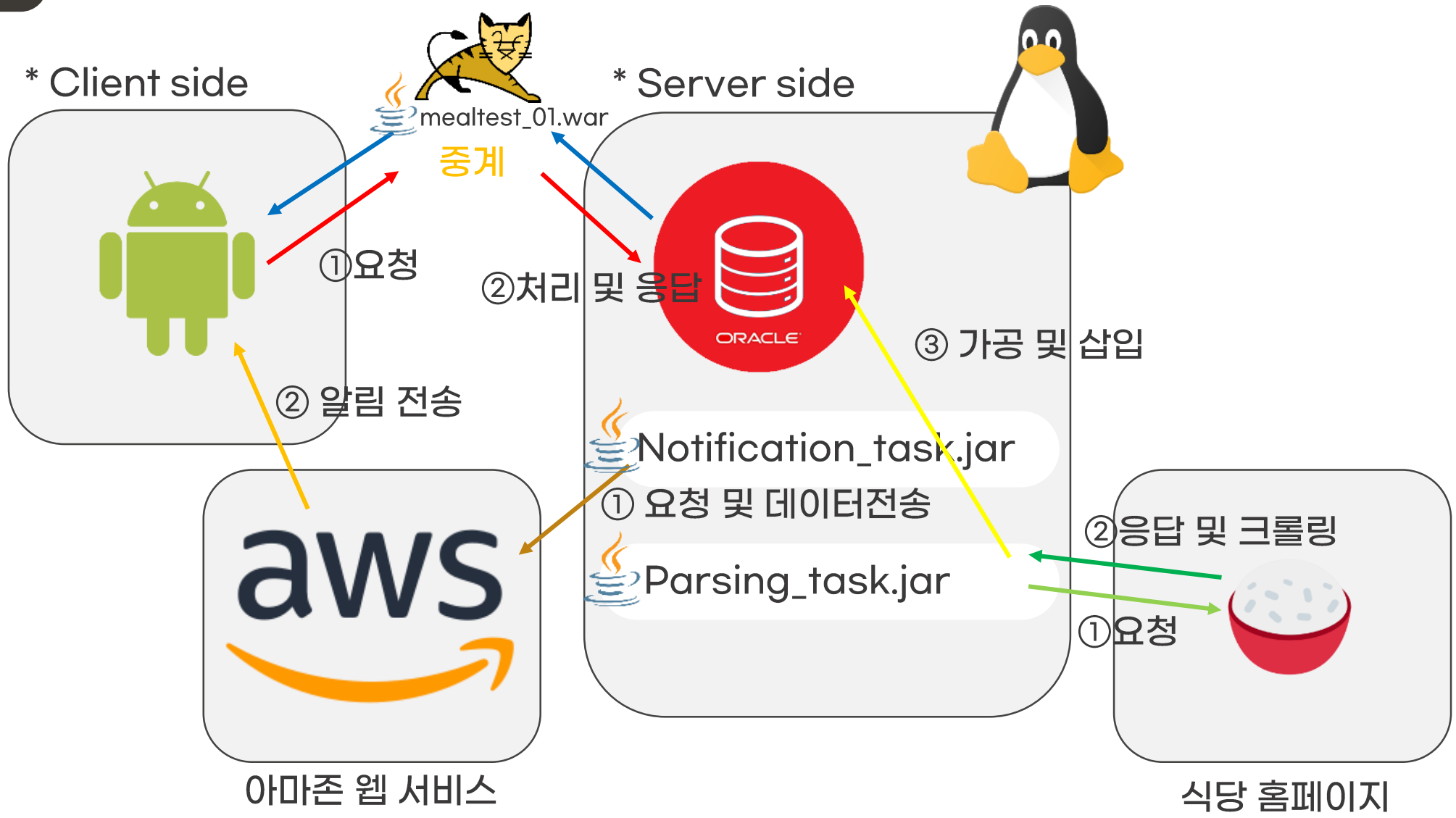
ORACLE 11g

TOMCAT

AMAZON WEB SERVICE

Part 2

개발 환경 - 클라이언트 및 서버 설계

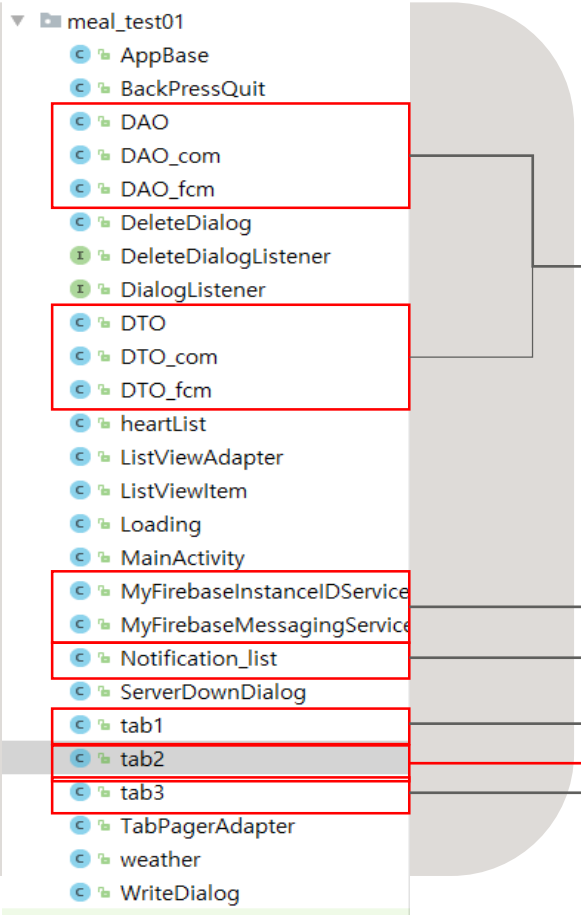


Part 3 소스 리뷰 및 문제해결

Part 3

소스 리뷰 및 문제해결 - 클라이언트 클래스 구성

- 클라이언트로 부터 JAVA에서 생성한 URL으로의 접근 과정에서, 제작할 시기 당시에 http통신을 이용하기위해서는 AsyncTask를 통해 비동기작업으로 처리해야 되었기 때문에 사용하였고, 현재는 deprecated되었으며, 코틀린의 OkHttp, Retrofit 등 새로운 기능을 이용해야 합니다.



DB로부터 데이터를 받아오는 클래스

*식당 메뉴 클래스.

FCM 서비스 이용하는 클래스.

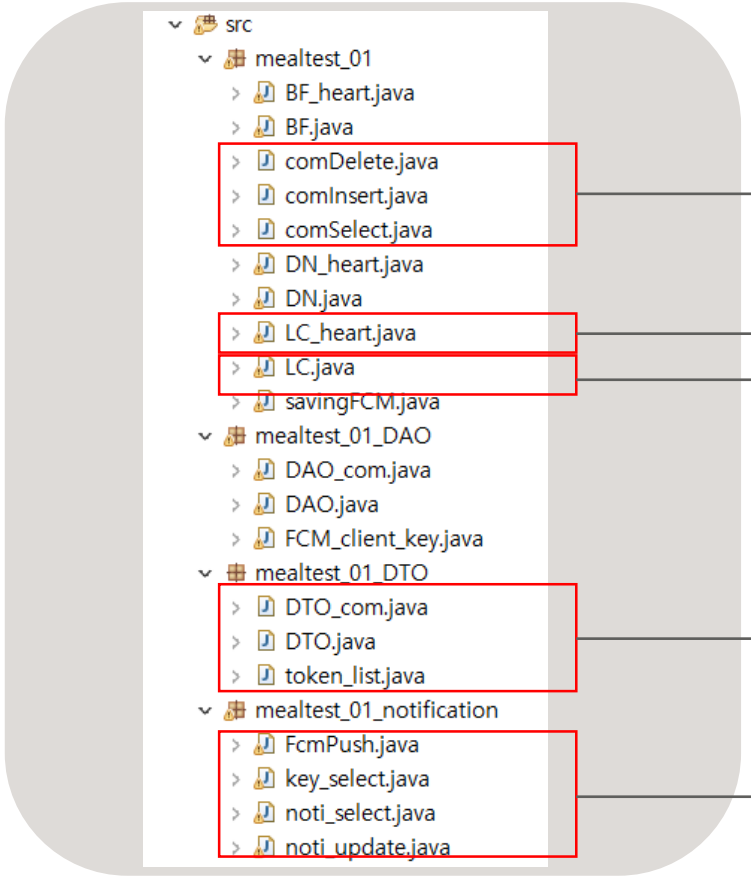
알림 기능을 켜놓은 유저 리스트 클래스.

공지 및 학교 인근의 날씨정보 클래스.

식사 후기 및 커뮤니케이션 클래스.

Part 3 소스 리뷰 및 문제해결 - meatest_01.war 클래스 구성

* DB서버(oracle) - 클라이언트(android studio) 간의 데이터 중계 구간입니다.
안드로이드 스튜디오에서 DB로의 직접적인 접근이 불가하여, java로 만든 URL로 접근 후,
오라클 서버에 접속하여 쿼리 명령문을 처리하였습니다.



후기 관련 DB 연결(삭제, 삽입, 조회) 클래스

선호도 관련 DB 조회 클래스

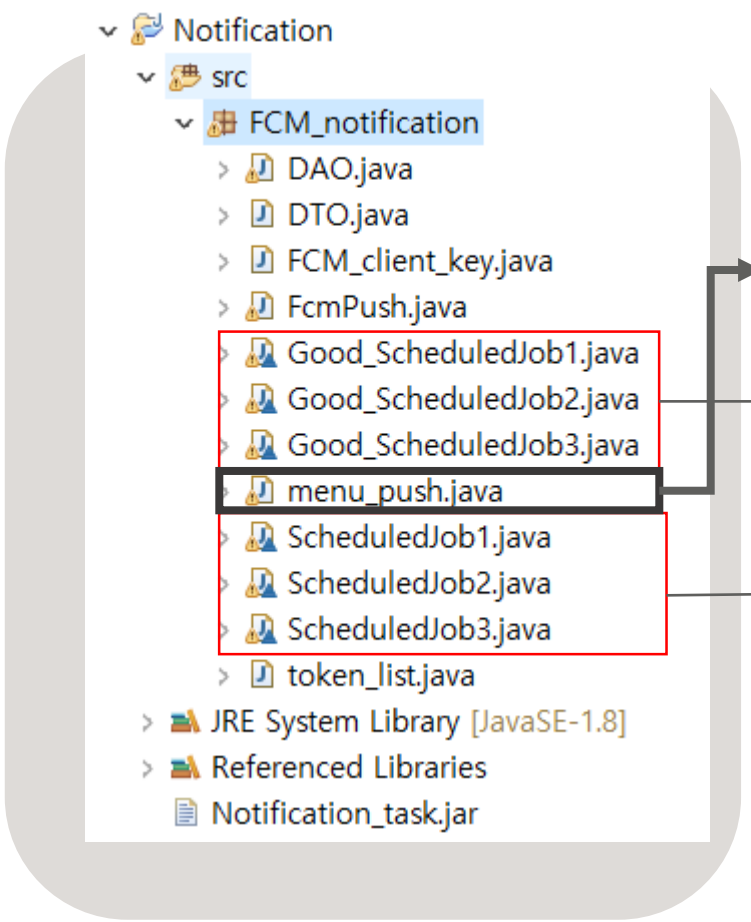
식단 관련 DB 조회 클래스

각 기능의 기본 데이터 셋

알림 기능 셋팅 관련 DB 연결 클래스

Part 3 소스 리뷰 및 문제해결 - Notification_task.jar 클래스 구성

* 메뉴 알림 기능은 선호도가 많은 메뉴/전체 메뉴, 그리고 본인이 아침/점심/저녁 중 선택한 시간을 반영하여 해당 식사시간전에 동작 할 수 있도록 설계하였습니다. 고유 user key와, 메시지를 FCM서버에 보내면 FCM서버로부터 각 클라이언트에 내용이 담긴 알림을 전송 하게 됩니다.



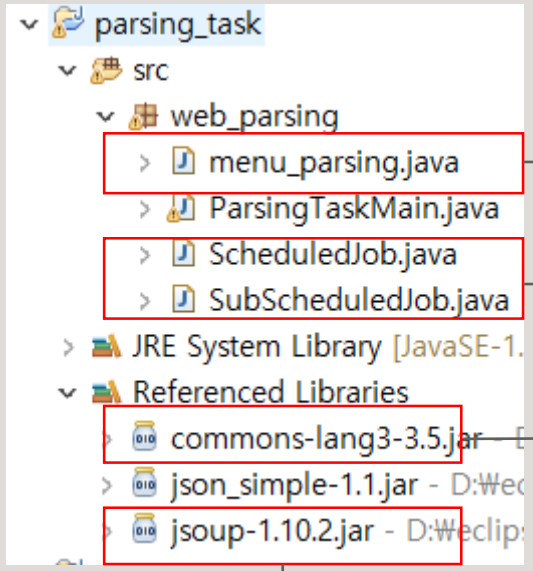
스케줄링(예약작업) 처리

- 아침/점심/저녁 각각의 알림 메세지 가공 및 FCM 서버로 메시지 전송 (선호도가 높게 나온 식단에 대해)
- 아침/점심/저녁 각각의 알림 메세지 가공 및 FCM 서버로 메시지 전송

Part 3

소스 리뷰 및 문제해결 - Parsing_task.jar 클래스 구성

* 학교 식당 홈페이지 내의 식단표를 jsoup 라이브러리를 사용하여, 해당 테이블에 있는 내용을 파싱 및 DB에 저장 시키는 작업을 자바 스케줄링을 통해 동작하도록 만들었습니다.



파싱 관련 메소드 정의

스케줄링 작업. 메인/서브로 나누어 메인 작업이 실패할 경우, 서브 스케줄링 작동

크롤링한 내용을 패턴으로 잘라 가공 할 수 있게 해주는 라이브러리

웹사이트에 접근하여 크롤링 할 수 있도록 해주는 라이브러리

Part 3 소스 리뷰 및 문제해결 - 문제점 리스트

첫 째	스케줄링 문제
둘 째	HTTP 통신 시 문제
셋 째	FCM KEY 기능

Part 3 소스 리뷰 및 문제해결 - 스케줄링 문제해결

```java
public class ScheduledJob extends TimerTask{
    public void run() {
        System.out.println("MAIN task start!!");
        try {
            Thread.sleep(20000);
        } catch (InterruptedException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
        Calendar now = Calendar.getInstance();
        System.out.println(now.getTime());

        Calendar checkcal = Calendar.getInstance();
        checkcal.set(Calendar.DAY_OF_WEEK, Calendar.MONDAY);

        menu_parsing mp = new menu_parsing();
        mp.delete_menu();
        mp.parsing();

        java.text.SimpleDateFormat fm = new java.text.SimpleDateFormat("yyyyMMdd");
        String MON = fm.format(checkcal.getTime());

        Timer subScheduler = new Timer();

        if(mp.check_DATA(MON)) {
            SubScheduledJob job2 = new SubScheduledJob();
            subScheduler.scheduleAtFixedRate(job2, 0, 1000*60*60*6); //6시간마다
            cancel();
        }
```

```
root      8113     1  0  3월 19 ?        00:40:05 java -jar Notification_task.jar
root      5770     1  0  4월 29 ?        00:09:02 java -jar parsing_tesk.jar
```

문제점

리눅스 기능 중 CRON을 사용하여
소모되는 자원을 최소화하고 싶었으나,

홈페이지의 식단표 업데이트 되는 시간이
매번 달랐기때문에 한주가 넘어가도
월요일 식단표가 업데이트 되지 않은 날도 있어,

한주가 넘어 갈 때, 월요일 데이터가 없는 경우에는
6시간마다 주기적으로 예약 작업을 해야 했습니다.

해결

자바의 스케줄링 작업 기능인 TimerTask 기능을 이용하여 작업을 조건에 따라 메인/서브로 나누어 실행되게 만들었습니다.

리눅스 서버에서는 해당 jar파일을 nohub 명령어로 백그라운드에서 작업이 진행되도록 처리하였습니다.

Part 3 소스 리뷰 및 문제해결 - 클라이언트 HTTP 통신시 문제(1)

서버 meatest_01.war 중 아침식사 데이터를 DB로부터 보내주는 소스

```
response.setCharacterEncoding("UTF-8");
DAO dao = new DAO();
ArrayList<DTO> list = dao.select1();
JSONObject jsonMain = new JSONObject(); // ??ü
JSONArray jArray = new JSONArray(); // ?迭
JSONObject jObject = new JSONObject(); // JSON?????? ???? ??ü.

for(DTO dto : list) {
    int i=0;
    today = dto.getToday();
    rice = dto.getRice();
    soup = dto.getSoup();
    side1 = dto.getSide1();
    side2 = dto.getSide2();
    side3 = dto.getSide3();
    side4 = dto.getSide4();
    dessert = dto.getDessert();
    good = dto.getGood();

    System.out.println("test2 :"+ today+','+rice+','+soup+','+side

    jObject.put("today", today);
    jObject.put("rice", rice);
    jObject.put("soup", soup);
    jObject.put("side1", side1);
    jObject.put("side2", side2);
    jObject.put("side3", side3);
    jObject.put("side4", side4);
    jObject.put("dessert", dessert);
    jObject.put("good", good);

    jArray.add(jObject.clone()); // ?????ë??_? ?Ƽ?? ?
}
```

클라이언트에서 아침식사 데이터를 DB로부터 받는 소스

```
public ArrayList<DTO> BF()  {
    String line, page = "";
    try {

        URL url = new URL( spec: "http://" + ip + ":"+port+"/mealtest_01/BF"); //요청 URL을 입력
        conn = (HttpURLConnection) url.openConnection();
        conn.setConnectTimeout(TIME_OUT_LANGTH);
        conn.setReadTimeout(TIME_OUT_LANGTH);
        conn.setRequestMethod("POST"); //요청 방식을 설정 (default : GET)
        conn.setDoInput(true); //input을 사용하도록 설정 (default : true)
        conn.setDoOutput(true); //output을 사용하도록 설정 (default : false)


        Log.d( tag: "loading",  msg: "getload");
        BufferedReader br = new BufferedReader(new InputStreamReader(conn.getInputStream(),  charsetName: "UTF-8")); //캐릭터셋
        while ((line = br.readLine()) != null) {
            page += line;
        }
        Log.d( tag: "receiveMsg : ", page);
        JSONObject json = new JSONObject(page);
        JSONArray jArr = json.getJSONArray( name: "BF");
    for (int i = 0; i < jArr.length(); i++) {
        json = jArr.getJSONObject(i);
        today = json.getString( name: "today");
        rice = json.getString( name: "rice");
        soup = json.getString( name: "soup");
        side1 = json.getString( name: "side1");
        side2 = json.getString( name: "side2");
        side3 = json.getString( name: "side3");
        side4 = json.getString( name: "side4");
        dessert = json.getString( name: "dessert");
        good = json.getString( name: "good");

        BFlist.add(new DTO(today, rice, soup, side1, side2, side3, side4, dessert, good,  flag: 0));
    }
```

Part 3 소스 리뷰 및 문제해결 - 클라이언트 HTTP 통신시 문제(2)

```
@Override
public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState) {
    v = inflater.inflate(R.layout.tab2, container,  attachToRoot: false);

    AsyncTask2 task2 = new AsyncTask2();
    task2.execute();


    return v;

}

class AsyncTask2 extends AsyncTask<Object, Void, Void>{
    LinearLayout ntp1,ntp2,ntp3;
    String key;
    TextView text1, text2, text3;
    TextView date;
    DAO_fcm dao_fcm;
    DTO_fcm noti;
    ImageButton back, next;
    ShineButton heart1, heart2, heart3;
    TextView gtext1, gtext2, gtext3;
    int count;
    int MAX = 6, MIN = 0;
    Calendar cal = Calendar.getInstance();
    @Override
    protected void onPreExecute() {
    }

    @Override
    protected Void doInBackground(Object... objects)  {
        super.onPreExecute();
        count = cal.get(Calendar.DAY_OF_WEEK)-1;

        dao = new DAO();
        dao_fcm = new DAO_fcm();
```

문제점

1. 클라이언트와 DB를 바로 연결시키지 못하고, 사이드 스크립트에서 DB데이터를 중계하여 데이터를 받아야 했습니다.

2. 안드로이드에서 http 통신을 하기위해서는 비동기 통신을 해야했습니다.

해결

1. 서브릿을 이용, Json 형식으로 가공하여 post방식으로 전송 하였고, 받을 때에도 Json형식으로 받았습니다. 이때, 인코딩에 주의해야합니다.

2. Asynctask를 이용하여 백그라운드에서 비동기 통신하였고, 현재는 deprecated 된 상태입니다.

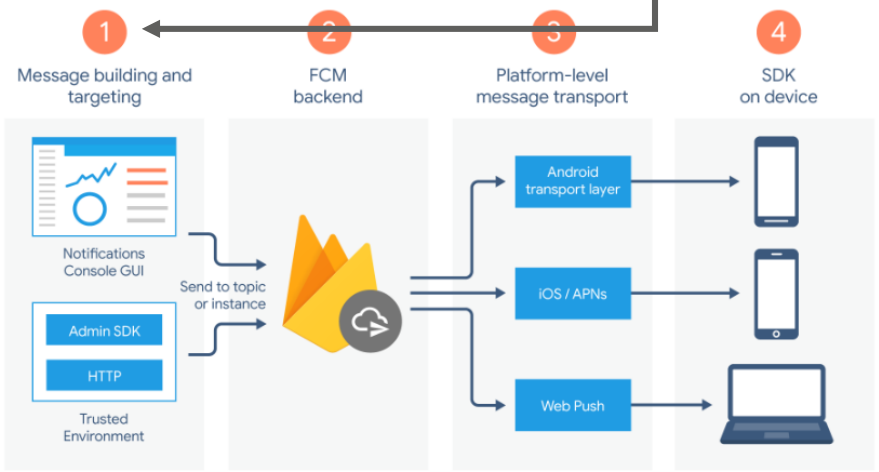
Part 3 소스 리뷰 및 문제해결 - FCM기능

```
lunch = dao.LC();
dinner = dao.DN();
try {
    SharedPreferences pref = getContext().getSharedPreferences(
            getString(R.string.client_key), MODE_PRIVATE);

    key= pref.getString( key: "token", defValue: "UnLoaded");
    Log.d( tag: "key",key);

    noti = dao_fcm.noti_select(key);

    if (noti.getToken() == null) {
        dao_fcm.key_insert(key);
        noti = dao_fcm.noti_select(key);
    }
```



문제점

기기마다 FCM KEY를 등록 하여야 알림서비스를 이용 할 수 있었고, 또 식사시간 마다 알림서비스를 따로 만들어야해서 로직을 어떻게 짤지 고민이 많이 됐습니다.

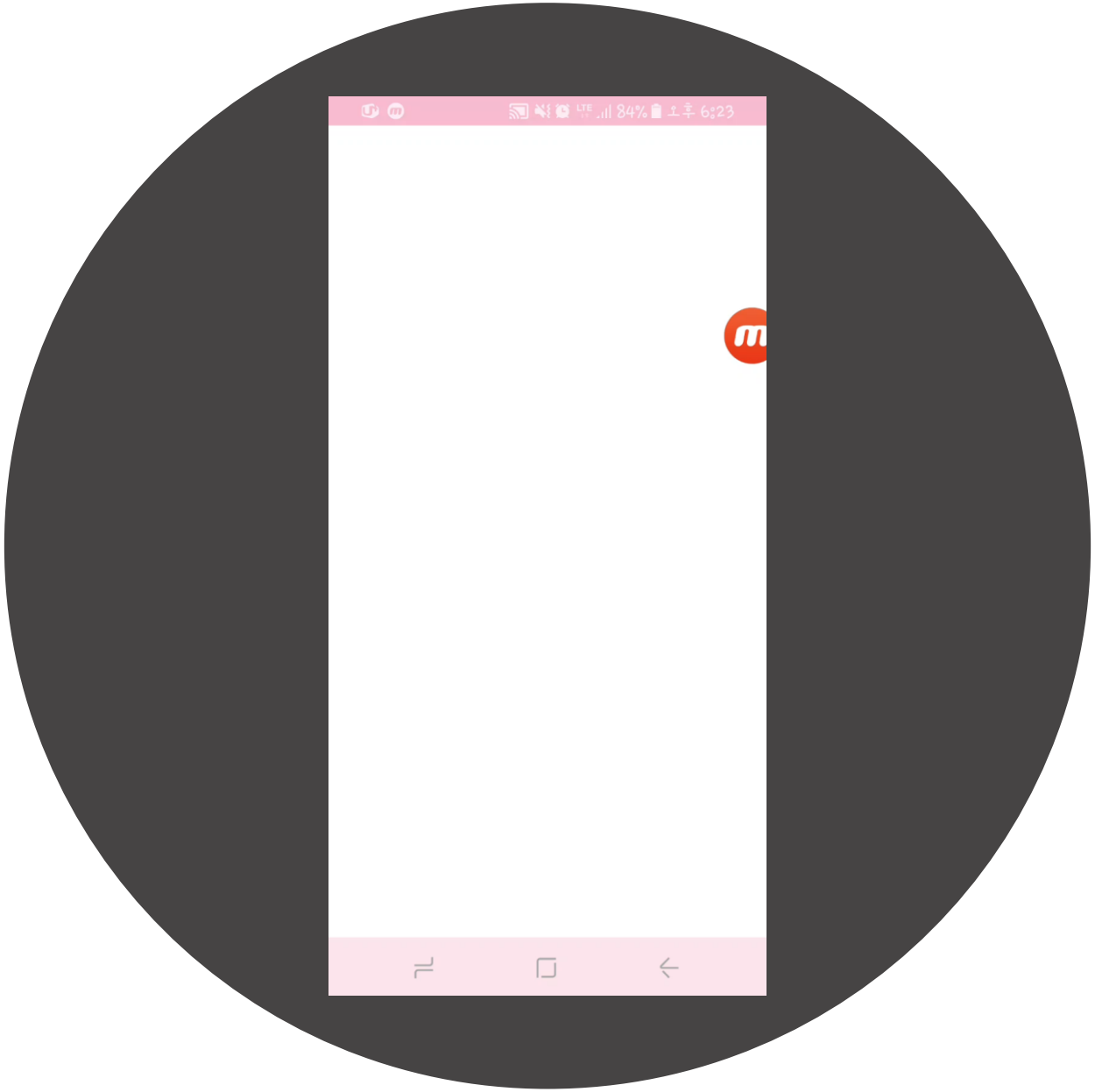
해결

SharedPreference 기능을 이용하면 클라이언트 로컬에 데이터를 저장할 수 있다는 걸 알게되었습니다. 최초 어플 실행 시 KEY값이 있는 지 확인 후 키 값이 없으면, FCM서버에 KEY값을 등록 하고 키 값을 로컬과 DB에 저장하도록 설계하였습니다.
또, 알림(on/off) 데이터를 키 값과 같은 테이블에 저장하여, 아침, 점심, 저녁 알림데이터를 함께 읽어올 수 있게 하였습니다.

Part 4 시연영상

Part 4

시연 및 정리 - 시연 영상



감사합니다