

# 소프트웨어공학적인 방법론을 고려한 소셜 네트워크 서비스의 그룹 관리용 어플리케이션의 설계 및 구현

이 상 곤\*

## Design and Implementation of a Effective Group Manager Application

Sangkon Lee\*

### 요 약

본 논문에서는 그룹 전용으로 사용하기 위한 소셜 네트워크 서비스(SNS)를 제안한다. 본 논문의 시스템 개발을 위해 그룹 서비스, 채팅, 위치 추적, 전송 기능, 그림 그리기와 편집 기능, 사진첩 관리 기능, 파일 전송 기능 등을 구현하고 전체 모듈을 결합하여 안드로이드용 그룹 관리 어플리케이션 시스템을 설계하였다.

### ABSTRACT

This paper is to present a new social network system that is convenient to group message system. For this system, we implement a group messaging system, chatting, position monitoring, file transfer, picture drawing and/or editing, photo management system, and we combine them as one group manager application based on smart system.

**Keywords** : Social Network System, Google Map, GPS, Smart Phone App, Software Engineering

## 1. 서 론

대부분의 사람들은 작은 소모임, 동기들, 동아리 모임 등의 여러 사회 활동을 하고 있다. 현대인은 여러 모임 약속을 위해 전화하거나, 단체 카카오톡, 단체 문자 등을 이용하는데, 그룹원들 간에 일정 조율이 쉽지 않고, 때로는 의사소통에 제약이 있다. 본 논문에서는 단체 모임 위해 필요한 "그룹 전용으로 사용 가능한 소셜 네트워크 서비스(SNS)"를 제작하고자 한다. 본 시스템의 완성을 위해 필요 기능을 나열하면 다음과 같은 점이 필요하다.

- 그룹원의 채팅,
- 그룹원의 사진첩,
- 그룹원의 연락처,
- 그룹원의 세부 일정 관리,
- 그룹원의 이동 상황을 보여주는 현재 위치 정보,
- 그룹원의 현재 위치에 대한 정보의 전송

본 논문의 목표는 다음과 같다. 본 시스템은 안드로이드용 어플리케이션으로 최근 스마트 폰의 사용량이 급격히 증가하며 점점 두터워지고 있는 사용자 그룹을 공략하는 스마트폰용 프로그램이다. 현재에는 페이스 북(facebook), 트위터(twitter), 카

\*교신저자: 진주대학교 컴퓨터공학과 언어과학연구소(samuel@jj.ac.kr),

접수일자 : 2013년 04월 29일, 수정일자 : 2013년 05월 12일, 심사완료일자 : 2013년 5월 25일

표 1. JSP를 이용한 DB 통신 방법의 구현 예  
Table 1. Implementation Example of DB Communication

```

public InputStream requestWriteAddressInvite(String requestURL,
String groupnumber, String phonenumber) {
    try {
        HttpClient client = new DefaultHttpClient();
        List<NameValuePair> dataList = new ArrayList<Name
ValuePair>();
        dataList.add(new BasicNameValuePair(
"groupnumber", groupnumber));
        dataList.add(new BasicNameValuePair(
"phonenumber", phonenumber));
        requestURL = requestURL + "?"
+ URLEncoderUtils.format(dataList, "UTF-8");
        HttpGet get = new HttpGet(requestURL);
        HttpResponse response = client.execute(get);
        HttpEntity entity = response.getEntity();
        InputStream is = entity.getContent();
    }
}
/* 일부 코드는 생략함 */
    
```

카오톡(Kakao Talk) 등 각 메신저들의 소셜 네트워크(social network) 서비스가 그 위세를 떨치고 있다. 본 논문을 통해 소셜 네트워크의 새로운 시장을 개척하고자 한다. 많은 사용자들에게 본 어플리케이션을 사용하고, 기능적/성능면에서 보다 안정된 서비스를 제공하여야 한다. 본 논문의 시스템 개발 시 나타나는 여러 제약 사항은 다음과 같다. 하드웨어 사양은 본 어플리케이션은 안드로이드 2.3 버전 진저 브레이드 이상에서 동작하며, 그 이하의 버전에서는 사용할 수 없다. 또한 제작 및 테스트를 위해 최소한 3대 이상 개통된 안드로이드 전용 스마트 폰이 필요하다.

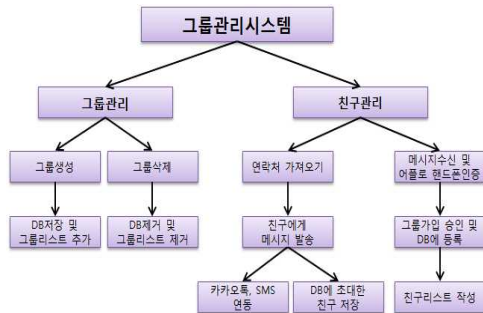


그림 1. 그룹 관리와 그룹원 관리 모듈의 구조도  
Fig. 1 Diagram of Group Manager

본 논문의 구성은 다음과 같다. 2장에서는 본 논문에서 제안하는 기능 및 요구 분석을 시스템 구조도와 함께 제시한다. 3장에서는 구현 기술로

JSP를 이용하여 데이터베이스의 통신과 XML Parsing 방법을 소개하고, SQLite의 데이터베이스 통신과 GPS 서비스의 구현 방법을 소개한다. 4장에서는 유스케이스 다이어그램과 명세서를 그리고 소프트웨어 공학에서 중요하게 다루는 시퀀스 다이어그램 클래스 다이어그램을 소개하고, 마지막으로 결론 및 추후 개선 사항에 대해 토의한다. 그리고 부록에 전체 모듈을 제시하였다.

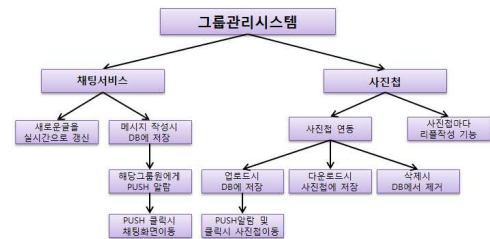


그림 2. 채팅 서비스와 사진첩 모듈의 구조도  
Fig. 2 Diagram of Chatter and Photo Manager

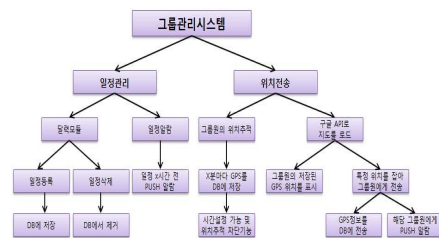


그림 3. 일정 관리와 위치 전송 모듈의 구조  
Fig. 3 Diagram of Memo and Location Manager

표 2. XML 파서의 구현 예  
Table 2. Implementation Example of XML Parser

```
<?xml version="1.0" encoding="utf-8" standalone="no" ?>
<%@page import="com.parser.*"%>
<%@page
    import="java.io.*, java.sql.*, java.util.*, java.net.*,
    java.io.BufferedReader"
    contentType="text/xml; charset=utf-8"%>

<addressInvite>
<%
    String getGroupnumber = request.getParameter("groupnumber");
    String getPhonenumber = request.getParameter("phonenumber");
    Address obj = new Address();
    getPhonenumber = obj.extractNumber(getPhonenumber);
    obj.writeAddressInvite(getGroupnumber, getPhonenumber);
%>
</addressInvite>
```

## II. 고객의 요구 분석

먼저 본 논문의 이론적 배경을 설명하기 위해 시나리오를 다음과 같이 설정하였다. 직장인 K씨는 고등학교 친구 5명과 함께 정기적으로 모임을 갖는다. 서로 약속을 정하거나 모임 이야기를 주고받는데 불편함을 느꼈다. 바쁜 L씨가 시간을 정해 팀원 모두에게 전체 문자를 보냈더니, P씨와 M씨가 시간이 안 되는 것이었다. 다시 모두에게 연락하여 가능한 시간을 조율하다 보니, 많은 시간이 오래 걸리고 여간 불편한 것이 아니었다. 이에 K씨는 소그룹을 관리하는 어플리케이션이 있다는 이야기를 듣고 이를 설치한다. K씨는 어플리케이션을 설치하고 그룹을 생성한다. 생성 후, 연락처에 있는 5명의 친구에게 가입 초대 카카오톡(or 문자) 메시지를 발송한다. 메시지를 확인한 친구들은 수락을 하고, 안드로이드 마켓 설치 페이지로 이동된다. 설치가 완료되면 핸드폰 번호로 인증하고 그룹 가입이 완료된다. 가입된 그룹원은 [연락처] 메뉴에서 확인하고 전화, 문자, 카카오톡 메시지를 발송한다. [채팅] 메뉴에서 서로 실시간 채팅을 이용한다. L씨가 찍은 셀카<sup>1)</sup> 사진을 사진

첩에 올렸고, 실시간으로 다른 친구들에게 푸시(PUSH)로 알람이 간다. 사진을 확인한 친구들은 리플로 사진에 대한 생각을 서로 나눈다. [일정] 메뉴에서 몇 일/ 몇 시에 모임을 가질 것인지 등록하고, 각각의 그룹원들은 약속 X 시간 전에 푸시 알람을 받을 수 있는 기능으로 약속에 참석한다. 약속은 잡았는데 약속 장소를 정확히 설명하기 어려운 경우가 있는데, 이 때는 '위치 전송' 기능을 통해 특정한 위치를 친구에게 전송하여 약속 장소에 찾아올 수 있도록 한다. 또한 친구들의 위치를 실시간으로 확인할 수 있어 편리하게 이용할 수 있다.

본 논문의 어플리케이션이 실행되면 자신의 핸드폰 번호로 인증을 받는다. 인증 후, 그룹 목록이 나오고 자신만의 그룹을 개설한다. 개설 후, 자신의 연락처로 친구들을 초대할 수 있다. 초대하면 문자나 카카오톡으로 초대 메시지가 발송된다. 초대를 받은 사용자들은 메시지의 링크를 선택하여 마켓의 어플리케이션 다운로드 페이지로 이동한다. 설치 후, 핸드폰 번호로 인증하고 자동으로 그룹에 추가된다. 데이터베이스에 그룹명과 핸드폰 번호가 암호화 되어 저장된다.

1) '셀프카메라(self camera)'를 줄여 이르는 말. 동아일보의 2004년 06월 18일 자 신문 기사에 다음의 문장에서 처음

출현함. 셀카가 유행처럼 번지는 것은 '잘 찍히는 것'에 대한 관심이 얼마나 많은지 엿볼 수 있는 사례다.

표 3. 자바 클래스에서 데이터베이스 연동을 구현 예  
Table 3. Implementation Example of Database in Java Class

```

public Address() {
    try {
        Class.forName("org.gjt.mm.mysql.Driver");
        con = DriverManager.getConnection(url, userid, passwd);
    } catch (Exception e) {
        e.printStackTrace();
    }
}

public void writeAddressInvite(String groupnumber,
String phonenumber) {
    try {
        String sql = "INSERT INTO GROUPINVITE
                    (GROUPNUMBER, PHONENUMBER) VALUES(?, ?)";
        stmt = con.prepareStatement(sql);
        stmt.setString(1, groupnumber);
        stmt.setString(2, phonenumber);
        stmt.executeUpdate();
    } catch (Exception e) {
        e.printStackTrace();
    }
    finally {
        try {
            stmt.close();
            con.close();
        } catch (SQLException se) {
            se.printStackTrace();
        }
    }
}
}

```

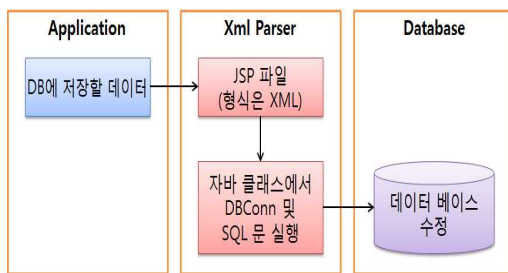


그림 4. 응용 프로그램과 DB의 통신 방법  
Fig. 4 Sequence Diagram between Application and DB Communication

그룹 네트워크의 구성과 친구 등록 과정은 다음과 같이 설계하였다. 그룹원들이 추가될 때마다 친구가 등록되고, [연락처] 메뉴에서 친구들을 확인할 수 있다. 등록된 친구들에게 전화, 문자, 카카오톡의 메시지 전송 기능을 할 수 있다. 그룹 네트워크를 통해 채팅, 사진첩, 일정 등 다른 서비스를 이용하기 위해 데이터가 전송되도록 설계한다. 데이터는 데이터베이스 서버에 저장되며, 어플리케이션을 켤 때 동기화 하여 핸드폰 데이터베이스에 저장된다. 그룹원들은 실시간으로 채팅 서비스를 이용할 수 있다. 한 명의 그룹원이 채팅을 입력하면 데이터베이스 서버에 전송되고

표 4. XML 파싱 방법의 구현 예

Table 4. Implementation Example of XML Parsing

```

/* 코드 일부 생략 */
    StrictMode.ThreadPolicy policy = new StrictMode.ThreadPolicy.Builder()
        .permitAll().build();
    StrictMode.setThreadPolicy(policy);
    String xml = "http://주소/jspXmlParsing/readAddressUserList.jsp?count="+ groupNumber;
    URL URL = new URL(xml);
    InputStream in = URL.openStream();
    XmlPullParserFactory factory = XmlPullParserFactory.newInstance();
    XmlPullParser parser = factory.newPullParser();
    parser.setInput(in, "utf-8");
    int eventType = parser.getEventType();
    while (eventType != XmlPullParser.END_DOCUMENT) {
        switch (eventType) {
            case XmlPullParser.START_DOCUMENT:
                break;
            case XmlPullParser.END_DOCUMENT:
                break;
            case XmlPullParser.START_TAG:
                if (parser.getName().equals("name")) {
                    inname = true;
                } else if (parser.getName().equals("number")) {
                    innumber = true;
                } else if (parser.getName().equals("date")) {
                    indate = true;
                }
                break;
            case XmlPullParser.END_TAG:
                break;
            case XmlPullParser.TEXT:
                if (inname) {
                    Name = parser.getText();
                    inname = false;
                } else if (innumber) {
                    Number = parser.getText();
                    innumber = false;
                } else if (indate) {
                    Date = parser.getText();
                    indate = false;
                    userList.add(new UserData(Name, Number, Date));
                }
                break;
        }
        eventType = parser.next();
    }
/* 일부 코드는 생략함 */

```

표 5. SQLite에 데이터의 저장 예  
Table 5. Example of Data in SQLite

```

SQLiteDatabase db;
SQLiteHelper mDBHelper = new SQLiteHelper(액티비티명.this);
db = mDBHelper.getWritableDatabase();
String groupNum = groupList.get(position).getGroupcount();
db.execSQL("DROP TABLE IF EXISTS CURGROUP");
db.execSQL("CREATE TABLE CURGROUP(NUMBER TEXT)");
db.execSQL("INSERT INTO CURGROUP VALUES(" + groupNum + ")");
    
```

푸시를 통해 다른 그룹원들에게도 알람이 발생한다. 알람을 누르면 어플리케이션이 실행되고, 채팅 화면으로 이동한다. 환경 설정에서 알람 ON/OFF 할 수 있다.

그룹만의 사진첩이 있는데, 사진을 올릴 수 있다. 사진의 업로드는 [찍어서 올리기]와 [갤러리에 선택하기]가 있다. 업로드하게 되면 데이터베이스 서버에 저장되고, 다른 그룹원들에게 푸시 알람이 간다. 각각의 사진에는 댓글을 작성할 수 있다. 사진첩의 사진들은 다른 그룹원들이 내려받을 수 있다. 삭제 권한은 사진을 올린 그룹원에게만 있다.

그룹원들의 현재 위치를 실시간으로 확인할 수 있다. 이 기능은 그룹원이 약속 장소에 어느 정도 근접하여 있는지 파악할 수 있게 한다. 초기에 어플리케이션을 설치하고 위치 정보 제공에 대한 동의를 받는다. 일정 시간마다 그룹원들의 GPS 정보를 데이터베이스 서버에 전송하고, 위치 확인 메뉴를 선택하면 데이터베이스 서버의 최신 GPS를 가져와 지도에 표시한다. 특정 위치를 전송하는 기능은 자신의 위치가 아닌 지도상의 특정 위치 정보를 다른 그룹원에게 전송하는 기능이다. 데이터베이스 서버에 저장한 그룹원만 해당 위치 정보를 푸시를 통해 수신할 수 있다. 주요 시스템의 몇 가지 구조도는 위의 (그림 1)~(그림 3)과 같다.

### III. 구현 기술

#### 3.1 JSP를 이용한 데이터베이스의 통신 방법

먼저 본 논문의 방법은 그룹 관리용 어플리케이션이므로 여러 통신 모듈의 효율적 이용이 중요하다. 특히, (그림 4)와 같이 JSP를 이용하여 데이터베이스의 통신 기법을 설명하면 다음과 같다. 안드로이드 어플리케이션에서 데이터를 DB에 전송하기 위해서는 아래와 같은 과정이 필요하다. 친구를 초대하였을 때 호출되는 메소드의 예이다. 파라미터로 세 가지를 받는데, requestURL은 jsp의 주소, groupnumber은 그룹 번호, phonenumber은 핸드폰 번호를 의미한다. dataList라는 커스텀 List를 만들고, 데이터를 저장한다. URL에 데이터를 UTF-8 형태로 붙이고, HttpGet 형식이 이 URL을 파라미터로 준다. 다음으로 HttpResponse를 만들고, HttpClient 객체의 execute(get)을 호출하면 해당 URL에 파라미터를 붙여 전송을 완료한다. 앞의 <표 1>은 위의 과정을 구현한 예이다.

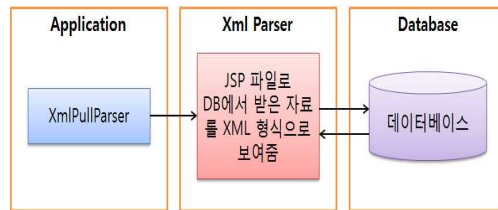


그림 5. JSP 이용 DB의 XML 파싱 방법  
Fig. 5 Sequence Diagram of XML Parsing

XmlParser의 동작에 대한 내용은 다음과 같다. writeAddressInvite.jsp 는 아래와 같은 코드로 구성되어 있다. 첫 번째 라인은 xml 형식을 의미하며, groupnumber 과 phonenumber 파라미터를 가

표 6. SQLite에서 데이터 호출 방법을 구현한 예  
Table 6. Implementation Example of Data Call Process in SQLite

```

Cursor cursor;
SQLiteDatabase db;
SQLiteHelper mDBHelper
    = new SQLiteHelper(액티비티
명.this);
db = mDBHelper.getReadableDatabase();
cursor = db.rawQuery("SELECT NUMBER
    FROM CURGROUP", null);
while (cursor.moveToNext()) {
    groupNumber = cursor.getString(0);
}
    
```

저오고, phonenumber의 형식을 맞추고, 다음으로 미리 만들어 둔 Address 클래스의 객체 obj를 만들어 writeAddressInvite (group number, phone number)를 호출한다. 앞의 <표 2>는 이에 대한 구현 예이다. JAVA 클래스에서 DB와의 연동 방법의 구현은 다음과 같다. writeAddressInvite(groupnumber, phonenumber)를 호출하면 Address에 있는 writeAddressInvite가 호출된다. 생성자로 MySQL 드라이버를 설정하고, writeAddressInvite 함수에서 SQL문을 작성하고, preparedStatement로 완성시킨 다음 executeUpdate()를 동작하여 DB의 내용을 변경한다. 앞의 <표 3>은 이에 대한 구현 예이다.

**3.2 JSP를 이용한 데이터베이스의 XML 파싱 방법**

먼저 어플리케이션에서의 동작은 다음의 (그림 5)와 같이 JSP를 이용한 데이터베이스의 XML Parsing을 설명한다. 어플리케이션에서 파싱을 위한 메소드는 <표 4>와 같다. XmlPullParser를 이용하여 주소와 파라미터를 입력 받고 파서를 세팅한다. while()을 수행하며, TAG 정보와 본문을 가져온다. 가져온 정보는 userList라는 어레이 리스트에 넣어 활용한다. 본 논문의 시스템이 XML에서 동작하도록 다음과 같이 하여야 한다. 먼저 readUserList.jsp에서의 동작이다. 파라미터로 count, 즉 그룹 번호를 받아온다. XML 형식의 파일을 작성하고, 데이터베이스를 접속하고, 완성된

SQL문을 실행하여 가져온 결과를 rs에 저장한다. rs.next()를 루프하며 resultSet에 저장하고 화면에 출력한다. JSP 파일이지만 모든 형식은 XML으로 작성된다. 이 과정의 구현 예를 [부록 B]에 제시하였다.

**3.3 SQLite 데이터베이스의 통신**

SQLite에서 데이터의 저장은 다음과 같다. 앱 프로그램을 만들 때는 어플리케이션 자체 DB를 사용하여야 하는 경우가 생기는데, 이 때 현재의 그룹 번호가 사용된다. CUR GROUP가 있다면 DROP 하고, 새로 CUR GROUP 테이블을 생성한다. 생성 후에 INSERT 하여 그룹 번호를 저장한다. 이것을 앞의 <표 5>와 같이 코드화 하였다.

**3.4 GPS 서비스의 구현**

표 7. 세부 GPS의 메소드  
Table 7. Details of GPS

```

locationMgr = (LocationManager) context.getSystemService(context.LOCATION_SERVICE);
// GPS가 켜져 있다면,
if (locationMgr.isProviderEnabled(LocationManager.GPS_PROVIDER)) {
    Criteria criteria = new Criteria();
    criteria.setAccuracy(Criteria.NO_REQUIREMENT);
    criteria.setPowerRequirement(Criteria.NO_REQUIREMENT);
    String best = locationMgr.getBestProvider(criteria, true);
    locationMgr.requestLocationUpdates(best, 1000, 0, this);
}
    
```

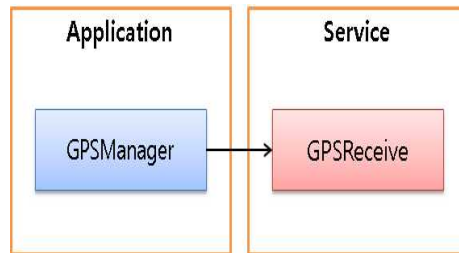


그림 6. & Fig. 6 GPS Service

다음으로 SQLite에서 데이터를 호출은 <표 6>

표 8. 위치 파악의 예  
Table 8. Example of Location

```

public void onLocationChanged(Location location) {
    if(bGetteringGPS == false){ // GPS를 1번만 조회
        double lat = location.getLatitude();
        double lon = location.getLongitude();
        Log.e("lat", lat+""");
        Log.e("lon", lon+""");
        Log.e("myPhoneNumber", myPhoneNumber);
        String requestURL =
            "http://주소:8080/jspXmlParsing/writeGPS.jsp";

        requestWriteGPS(requestURL, myPhoneNumber, String.
            valueOf(lat), String.valueOf(lon));
        bGetteringGPS = true;
    }
}
    
```

과 같다. 채팅, 일정 관리, 위치 추적, 위치 전송 등 시스템의 모든 기능을 사용할 때 필수적으로 기억해야 할 요소가 그룹 번호이다. 이 그룹 번호를 가져오기 위해 Cursor와 SQLite를 사용한다.

GPS Service의 구현은 (그림 6)과 같다. GPSReceiver를 서비스로 등록하는 방법은 다음과 같다. GPS 서 비스는 어플리케이션이 실행되자마자 동작한다. GPSManager() 메소드가 동작하고, 알람 매니저 서비스를 호출한다. setRepeating 할 때 GPSReceiver와 인터벌 10분을 파라미터로 준다. GPSReceiver 리시버가 알람 매니저 서비스에 등록되고, 10분마다 GPSReceiver가 호출된다. 이 방법의 구현 예를 [부록 C]에 제시하였다. GPSReceiver에서 위치의 추적 방법은 먼저 핸드폰의 GPS를 찾기 위해 locationManager 객체를 생성하고 등록한다. 만약 GPS가 꺼져 있다면 GPS를 전송하지 않고, 반면에 켜져 있다면 if문으로 들어가 아래의 위치를 추적하는 과정을 수행한다. 위의 <표 7>은 세부 GPS를 조회하는 메소드이다. <표 8>과 같이 double형으로 위도와 경도를 생성하고, location.getLatitude()와 location.getLongitude()를 통해 위도 경도를 파악하고, 찾은 위치 정보를 write GPS를 통해 데이터베이스 서버로 전송된다.

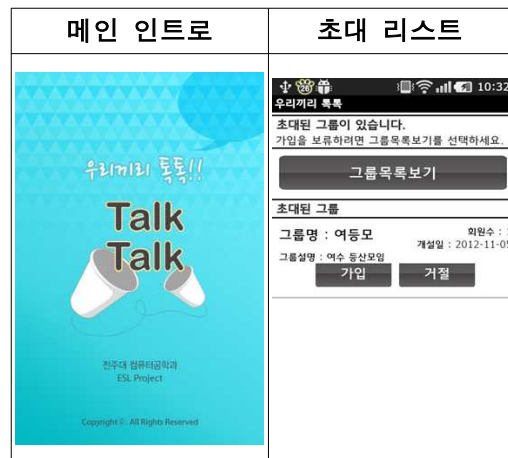


그림 7. 사용자 인터페이스  
Fig. 7 User Interface

(그림 7)과 같이 메인 화면을 디자인하였고, 초대 리스트의 수행 결과이다.



#### IV. 다이어그램을 이용한 코드의 이해

##### 4.1 유스케이스 명세서

시나리오 이름	그룹 관리
참 여 액 터	User에 의하여 구동됨
시 작 조 건	User가 어플리케이션을 실행하면 시작된다.
사건의 흐름	<p>* 정상 흐름</p> <ol style="list-style-type: none"> <li>1. User가 초대된 그룹이 있는지 DB에서 확인하여 있다면, 리스트 뷰로 그룹명, 그룹 회원수, 초대자, 메시지를 출력하고, [가입], [거절]을 표기한다.</li> <li>2. 가입시 닉네임을 입력받고, [가입 완료]버튼 클릭시 중복검사를 통해 가입을 완료한다. 거절시, 리스트에서 제거한다.</li> <li>3. User가 가입한 그룹들이 있다면, 화면에 리스트 뷰로 표기한다.</li> <li>4. [그룹 생성] 버튼으로 생성 화면으로 이동한다.</li> <li>5. 그룹명, 그룹 설명, 닉네임을 입력 받는다.</li> <li>6. 그룹명은 [중복 확인]을 해야 그룹을 생성할 수 있다.</li> <li>7. 그룹 생성이 완료되면 가입된 그룹을 리스트 뷰로 표기한다.</li> </ol>
종 료 조 건	User가 어플리케이션을 종료하거나, 그룹을 선택해 로그인하면 종료된다.

##### 4.2 사진첩의 사용 사례 명세서

일정 관리의 사용 사례 명세서와 위치 추적의 사용 사례 명세서, 위치 전송의 사용 사례 명세서는 생략한다.

시나리오 이름	사진첩
참 여 액 터	User에 의하여 구동됨
시 작 조 건	User가 [사진첩] 메뉴를 선택하면 시작된다.
사건의 흐름	<p>* 정상 흐름</p> <ol style="list-style-type: none"> <li>1. DB에서 사진첩의 개수를 가져와 어플리케이션의 사진첩 개수와 비교한다.</li> <li>2. DB의 개수에서 어플리케이션의 개수를 뺀 만큼을 DB에서 가져와 어플리케이션에 저장한다.</li> <li>3. 어플리케이션의 데이터를, 한 화면에 9개씩 가져와 그리드뷰 방식으로 출력한다.</li> <li>4. 사진을 선택하면 사진을 크게 보이고, 하단에 리플을 작성할 수 있다.</li> <li>5. DB에서 선택된 사진첩의 리플 개수를 가져와 어플리케이션의 선택된 사진첩의 리플 개수와 비교한다.</li> <li>6. DB의 개수에서 어플리케이션의 개수를 뺀 만큼을 DB에서 가져와 어플리케이션에 저장한다.</li> </ol> <p>* 선택 흐름</p> <ol style="list-style-type: none"> <li>7. 어플리케이션의 리플 데이터를 가져와 사진의 아래에 리스트뷰를 출력한다.</li> </ol> <p>- 리플을 작성하고 [입력] 하였다면, DB에 데이터를 저장하고, 정상 흐름을 5~7을 동작한다.</p>
종 료 조 건	User가 어플리케이션을 종료하거나, 다른 메뉴를 선택하면 종료된다.

친구 관리와 채팅 서비스의 사용 사례 명세서는 생략한다.

### 4.3 그룹 관리를 위한 시퀀스 다이어그램

그룹 관리의 시퀀스는 Group 클래스와 XmlParser 클래스가 참여한다. 프로그램이 시작되면, Group 클래스에서 XmlParser 클래스의 readInvite를 호출하여 초대된 그룹 목록을 받아와 inviteList를 이용하여 사용자에게 출력한다. 가입 시, 닉네임을 입력하면 XmlParser 클래스의 checkName을 호출하여 중복 확인을 하고 그룹 가입을 완료시킨다. 만약 가입을 거절하면, XmlParser 클래스의 delInvite가 동작하여 DB에서 초대한 데이터를 삭제하게 된다. 만약 초대된 그룹이 '0' 이면, XmlParser 클래스의 myGroup를 호출하여 자신이 가입된 그룹 목록을 가져와 사용자에게 출력한다. 만약 그룹 생성 버튼을 선택하면 XmlParser 클래스의 checkName을 호출하여 중복을 확인하고 그룹 생성을 완료한다. 이것은 (그림 8)과 같다.

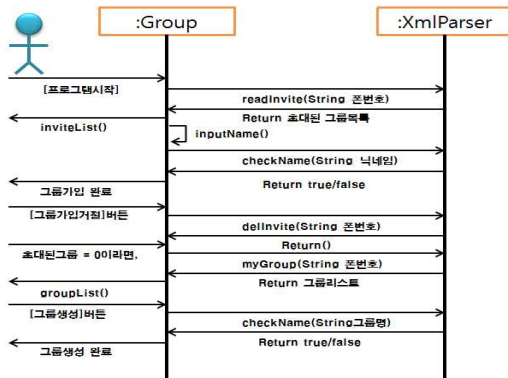


그림 8. 그룹 관리를 위한 시퀀스 다이어그램  
Fig. 8. Sequence Diagram for Group Manager

### 4.4 친구 관리

친구 관리의 시퀀스는 Address, XmlParser, AddressInvite 클래스가 참여한다. 이것은 주소록이 선택되었을 때 동작하며 Address 클래스에서 XmlParser 클래스의 readMyGroupList를 호출하여 자신의 그룹원 정보를 받아와 사용자에게 그룹원 목록을 출력한다. 초대하기 버튼을 선택하면 AddressInvite 클래스의 readMyAddress가 호출되어 자신의 연락처의 데이터를 화면에 출력하게 된다. 다음으로 초대할 사용자를 터치할 경우,

clickListener가 호출되면서 적절한 그룹원인지 확인하기 위해 Address 클래스의 checkGroupUser를 호출하여 중복 확인을 실시한다. 그룹원이 아니면 alertDialog로 다이얼로그를 호출하고, 초대하게 되면 sendSMS를 호출하여 선택된 대상에게 자동으로 SMS가 발송된다. 발송이 완료되면 XmlParser 클래스의 writeInvite가 호출되어 초대된 내역을 DB에 저장하게 된다.

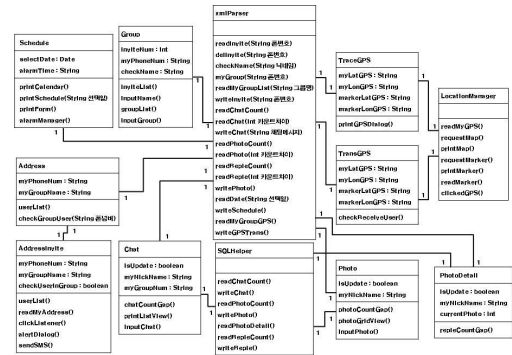


그림 9. 클래스 다이어그램  
Fig. 9. Class Diagram

### 4.5 채팅 서비스

친구 관리의 시퀀스는 Chat, XmlParser, SQLHelper 클래스가 참여한다. 채팅이 선택되면 동작하며 Chat 클래스에서 XmlParser 클래스의 readChatCount를 호출하여 DB의 채팅 개수를 가져온다. 같은 방법으로 SQLHelper 클래스의 readChatCount를 호출하여 어플리케이션의 채팅 개수를 가져온다. chatCountGap을 호출하여 업데이트 해야 할 채팅 개수를 확인하고, XmlParser 클래스의 readChat를 호출하여 업데이트 된 채팅 개수 만큼을 불러온다. 업데이트된 채팅을 SQLHelper 클래스의 writeChat를 호출하여 자신의 어플리케이션에 저장하고, printListView를 호출하여 화면에 출력한다. 만약, 사용자가 채팅을 입력할 경우 XmlParser 클래스의 writeChat를 호출하여 DB에 저장하고, 위의 채팅 업데이트 로직을 다시 수행한다.

### 4.6 사진첩

사진첩의 시퀀스는 Photo, XmlParser,

SQLHelper, PhotoDetail 클래스가 참여한다. 이 클래스는 사진첩이 선택되었을 때 동작하며 Photo 클래스에서 XmlParser 클래스의 readPhotoCount를 호출하여 DB의 사진첩 개수를 가져온다. 같은 방법으로 SQLHelper 클래스의 readPhotoCount를 호출하여 어플리케이션의 사진첩 개수를 가져온다. photoCountGap을 호출하여 업데이트 해야 할 사진첩 개수를 확인하고, XmlParser 클래스의 readPhoto를 호출하여 업데이트된 사진첩 개수 만큼 불러온다. 업데이트된 사진첩을 SQLHelper 클래스의 writePhoto를 호출하여 자신의 어플리케이션에 저장하고, photoGridView를 호출하여 화면에 출력한다.

만약 사용자가 사진을 클릭할 경우, PhotoDetail 클래스에서 SQLHelper 클래스의 readPhotoDetail를 호출하여 사진의 세부 정보를 화면에 출력한다. 다음으로 사진에 있는 리플도 출력해야 한다. XmlParser 클래스의 readRepleCount를 호출하여 DB의 해당 사진첩의 리플 개수를 가져온다. 같은 방법으로 SQLHelper 클래스의 readRepleCount를 호출하여 어플리케이션의 사진첩의 리플 개수를 가져온다. repleCountGap을 호출하여 업데이트해야 할 사진첩의 리플 개수를 확인하고, XmlParser 클래스의 readReple를 호출하여 업데이트 된 사진첩의 리플 개수 만큼을 불러온다. 업데이트 된 사진첩의 리플을 SQLHelper 클래스의 writeReple를 호출하여 자신의 어플리케이션에 저장하고 화면에 출력한다. 만약 리플을 작성하면 XmlParser 클래스를 이용하여 DB에 저장하고, 리플 업데이트 로직을 다시 호출한다.

#### 4.7 일정 관리

일정 관리의 시퀀스는 Schedule, XmlParser, 클래스가 참여한다. 일정 관리가 선택되었을 때 동작하며 Schedule 클래스에서 printCalendar를 호출하여 화면에 달력을 출력한다. 다음으로 XmlParser 클래스의 readDate를 호출하여 printSchedule를 이용하여 일정을 화면에 출력한다. 일정 입력 버튼을 선택하면 printForm을 이용하여 입력폼을 화면에 출력하고 입력 시, XmlParser 클래스의 writeSchedule를 호출하여

일정을 등록한다. 일정 입력이 모두 끝나고 alarmManager를 호출하여 알람을 등록하게 된다.

#### 4.8 위치 추적

위치 추적을 위한 시퀀스는 TraceGPS, XmlParser, LocationManager 클래스가 참여한다. 이 클래스는 위치 추적이 선택되었을 때 동작하며, TraceGPS 클래스에서 LocationManager 클래스의 readMyGPS를 호출하여 자신의 위치 정보를 가져오고, LocationManager 클래스의 requestMap를 호출하여 지도를 요청하면 printMap을 이용해서 구글 지도 API와 연동하고, 각종 지도에 대한 세팅한다. 지도의 화면은 자신의 좌표로 하여 설정하게 된다. 다음으로 TraceGPS 클래스에서 XmlParser 클래스의 readMyGroupGPS를 호출하여 그룹원들의 위치 정보를 전송 받는다. 받은 위치 정보를 LocationManager 클래스의 requestMarker를 호출하여 마커 등록을 요청한다. printMarker가 호출되어 지도에 그룹원들의 위치에 대한 마커가 등록된다. 만약 마커를 선택하면 LocationManager 클래스의 readMarker가 동작하여 마커에 대한 정보를 printGPSDialog를 이용하여 출력한다.

#### 4.9 위치 전송

위치 추적의 시퀀스는 TransGPS, XmlParser, LocationManager 등의 클래스가 참여한다. 이 클래스는 위치 전송이 선택되면 동작하며, TransGPS 클래스에서 LocationManager 클래스의 readMyGPS를 호출하여 자신의 위치 정보를 가져오고, LocationManager 클래스의 requestMap를 호출하여 지도를 요청하면 printMap을 이용해서 구글 지도 API와 연동하고, 각종 지도에 대한 세팅을 시작한다. 지도의 화면을 자신의 좌표로 설정한다.

만약 사용자가 지도의 특정 위치를 선택하면 LocationManager 클래스의 clickedGPS가 호출되어 선택된 좌표에 마커가 등록 된다. 위치 전송을 선택하면 checkReceiveUser가 호출되어 그룹원 중 받을 사람을 선택한다. 선택이 완료되고 전송 시 XmlParser 클래스의 writeGPSTrans가 호출되어 DB에 받을 사람과 좌표 등의 위치를 전송하

는 정보가 저장되고 최종적으로 전송 완료 메시지를 출력한다. 클래스 다이어그램은 다음의 (그림 9)와 같다.

#### IV. 결 론

본 논문에서는 안드로이드용 그룹 관리 어플리케이션 시스템을 설계하고, 그룹 서비스, 채팅, 위치 추적, 전송 기능, 그림 그리기와 편집 기능, 사진첩 관리 기능, 파일 전송 기능 등을 구현하고 전체 모듈을 결합하여 구글 플레이(구 마켓)에 '우리끼리 톡톡' 이라는 이름으로 업로드 되어 있으며, 여러 사용자가 실제로 사용하고 있다. 메신저 어플리케이션의 특성상 성능면에서 더 빠른 속도를 보장하여야 하며, 데이터베이스의 철저 관리로 데이터의 소실이나 유출이 차단한다. 또한 사용자의 피드백을 통해 지속적으로 프로그램을 버전 업 하여야 한다. 현재 그룹 관리, 친구 관리, 채팅 서비스, 일정 관리, 위치 추적, 위치 전송 등의 사용 사례를 구현하였으며, 사진첩과 푸시 알람 기능은 추후 개선 사항으로 진행될 것이다.

본 시스템은 최근 사용량이 급격히 증가한 스마트폰 유저들을 공략할 것이다. 페이스북(facebook), 트위터(twitter), 카카오톡(KakaoTalk) 등 각 메신저 들이 소셜 네트워크라는 서비스 아래 한창 위세를 떨치고 있는 중이다. 카카오톡의 약점인 그룹 기능을 잘 공략할 것으로 판단된다. 또한 어플리케이션을 통해 소셜 네트워크의 새로운 시장을 더 확대할 것으로 기대한다.

추후에는 사진첩과 푸시 알람 기능에 대한 수정이 필요하다. 현재 수정이 진행중이며 수정이 완료되면 구글 플레이 등을 통해 버전 업을 진행할 것이다. 또한 속도 최적화의 문제인데, 우선 일반 PC를 서버로 구성하였고, 웹 서버와 DB서버를 같은 컴퓨터에서 사용하고 있다. 추후에는 속도 개선을 위해 안정적인 인터넷 속도와 PC 사양을 보완하여 속도 개선에 최선을 다해야 할 것이다. 버그 수정은 다음과 같다. 일부 예외 처리되지 않은 항목들이 테스트 과정에서 발견되었으며, 코드 수정을 통해 버그가 발생하지 않도록 대비책을 마련할 것이다. 기타 편의 기능의 업그레이드는 친구를 초대할 때 카카오톡 메시지 등을 활용한다면 SMS 비용이 소요되지 않을 것이다.

[부록 A] 그룹 관리 어플리케이션의 전체 모듈 구성도



(부록 B) 데이터베이스의 XML 파싱 방법의 구현 예

```

<?xml version="1.0" encoding="utf-8" standalone="no" ?>
  try {
    Connection con = null;
    Statement stmt = null;
    String getCount = request.getParameter("count");
    String sql = "SELECT NAME, NUMBER, ADATE FROM GROUP" +
      getCount;
    ResultSet rs = null;
    String retResult = "<readAddressUserList>";
    try {
      Class.forName("org.gjt.mm.mysql.Driver");
      con = DriverManager.getConnection(
        "jdbc:mysql://localhost/groupmanager", "dblogin",
        "dddd");
      stmt = con.createStatement();
      rs = stmt.executeQuery(sql);
      while (rs.next()) {
        retResult += "<name>" + rs.getString(1) + "</name>";
        retResult += "<number>" + rs.getString(2)
          + "</number>";
        retResult += "<date>" + rs.getString(3) + "</date>";
      }
      retResult += "</readAddressUserList>";
      closeConn(rs, stmt, con);
      out.println(retResult);
    } catch (SQLException ex) {
      System.out.println(ex);
    }
  } catch (Exception ex) {
    System.out.println(ex);
  }
}
%>
<%!public void closeConn(ResultSet rs, Statement stmt, Connection con)
  throws Exception {
  rs.close();
  con.close();
  stmt.close();
}%>

```

(부록 C) GPS의 구현 예

```

public void GPSManager(){
  long firstTime = SystemClock.elapsedRealtime();
  firstTime += 1*1000;
  long interval = 1000*60*10;
  Intent intent = new Intent(this, GPSReceiver.class);
  pendingIntent = PendingIntent.getBroadcast(this, 0, intent, 0);
  AlarmManager am = (AlarmManager) getSystemService
    (ALARM_SERVICE);
  am.setRepeating(AlarmManager.ELAPSED_REALTIME_WAKEUP,
    firstTime, interval, pendingIntent);
}

```

## 참 고 문 헌

- [1] 안드로이드 정의 :  
<http://100.naver.com/100.nhn?docid=922356>
- [2] Open API : <http://terms.naver.com/entry.nhn?docId=20297>
- [3] 안드로이드 개발 레퍼런스 :  
<http://developer.android.com/reference/android/package-summary.html>

---

### 저자 약력

---

이 상 곤(Samuel Sangkon Lee)

정회원



1998년 전북대학교 전산통계학과  
(이학 석사)  
2001년 日本 도쿠시마대학교 지능  
정보공학과 (공학 박사)  
2002년 원광대학교 음성 정보 기술  
산업 지원 센터 연구원,  
2002년~현재 전주대학교 컴퓨터공  
학과 교수

<관심 분야> 한국어 정보 처리, 한글 공학, 정보검색,  
문서 분류 및 요약, 키워드 추출, 컴파일러,  
인공지능에 대한 연구