

SNS 기능을 적용한 안드로이드 웹하드 서비스 어플리케이션

신성욱*, 이승규*, 김진현*, 공기석*, 이상우**

*한국산업기술대학교 컴퓨터공학과

**유니챌 주식회사

e-mail : {mlvovlm°, fast365, evergogo, kskong}@kpu.ac.kr

swlee@unichal.com

Android Webhard Service App Applying SNS

Seong-Wook Shin^{o*}, Seung-Kyu Lee^{*}, Jin-Hyun Kim^{*}, Ki-Sok Kong^{*}, Sang-Woo Lee^{**}

^{*}Department of Computer Engineering, Korea Polytechnic University

^{**}Unichal, Inc.

요 약

클라우드 시스템이라고 할 수 있는 웹하드 서비스에 소셜 네트워크 서비스 기능을 적용한 안드로이드 어플리케이션이다. 일정한 웹하드 공간을 제공하고 사용자들은 파일 업로드, 다운로드, 파일공유 서비스를 제공받는다. 소셜 네트워크 서비스 기능을 적용함으로써 지인들과 사진, 음악, 동영상, 일반파일 등을 공유할 수 있고 이 파일들은 웹하드 서비스와 연동된다. 동영상의 경우 스트리밍 서비스를 제공해 실시간으로 시청이 가능하도록 개발하였다.

1. 서론

전 세계적으로 스마트폰 시장이 빠르게 급증하고 있는 최근, 안드로이드 OS의 확산은 빠르게 진행되고 있다. 오픈된 환경으로 큰 제약 없이 쉽게 개발이 용이하며, 다양한 스마트폰 제품들이 안드로이드를 채택하여 출시되고 있다.

스마트폰 이용률 증가에 따라 소셜 네트워크 서비스가 웹에서 뿐만 아니라 스마트폰에서의 사용이 증가되고 있다. 소셜 네트워크 서비스는 인맥형성에 큰 영향을 미치고 있으며, 기업에서 마케팅 전략으로 사용하는 등 사용자들의 삶에 큰 부분을 차지하고 있고 이용률 또한 꾸준히 증가하고 있다. 또한 인터넷 속도 증가 등 물리적인 네트워크 환경이 발전하면서 인터넷 접속이 가능한 곳에서는 언제든 사용할 수 있고 손쉽게 데이터를 공유할 수 있는 그래서 편리함을 더해주고 있는 클라우드 컴퓨팅이 많이 사용되고 있는 추세이다.

본 시스템은 클라우드 시스템에 SNS 기능을 적용하여 두 시스템의 장점을 살리고 기존 SNS 서비스와 차별성을 두어 좀 더 편리하고 유대 있고 스마트한 인간관계형성을 유도하자는 의미에서 구현 되었다. SNS 기능에 파일 공유 기능이 적용되어 사용자에게 편리함을 제공하고 공유한 파일을 토대로 사진, 동영상을 실시간으로 볼 수 있는 스트리밍 서비스를 제공한다[1][2].

2. 관련연구

2.1 기존 유사 App과의 비교

현재 안드로이드를 이용한 각종 SNS App들이 있다. 그 중에서 본 논문의 App과 유사한 기능을 지닌 App들은 '트위터', '페이스북', '카카오톡' 등이 있다. <표1>은 기존 SNS App인 '카카오톡', '트위터', '페이스북'과 해당 논문의 App과의 기능을 비교를 나타낸다.

<표1> 기존 SNS App과 논문 App의 SNS기능 비교

	카카오톡	트위터	페이스북	개발 App
동영상 스트리밍	O	O	O	O
파일공유	X	X	X	O
사진보기	O	O	O	O
전화번호부 친구추가 기능	O	X	X	O



카카오톡



트위터



페이스북

논문의 App은 현재 널리 사용되고 있는 SNS인 ‘트위터’, ‘페이스북’과 스마트폰 SNS의 대표적 App인 ‘카카오톡’의 장점을 살려 적용했다. 친구등록은 ‘카카오톡’에서 사용자 단말기의 전화번호부를 읽어와 친구를 추가 하는 방식을 적용하였고 콘텐츠 공유는 팔로어들 간에 쓴 글이나 사진 동영상 등을 공유하는 트위터의 방식을 적용 하였다. 클라우드 서비스를 이용 하면서 SNS를 함께 이용함으로써 사용자들 간 파일의 공유가 쉽고 편리해 진다는 것이 특징적이다. <표2>는 기존의 웹하드 App과 본 논문의 App의 웹하드 기능 비교를 나타내며 <표3>은 공통점과 차이점을 나타낸다.

<표2> 기존 웹하드 App과 논문 App의 웹하드기능 비교

	N드라이브	다음 클라우드	개발 App
파일공유	X	X	O
SNS	X	X	O
기존 App 이미지			

<표3> 기존 웹하드 App과 논문 App의 공통점 및 차이점

	기존 웹하드 App	개발 App
공통점	<ul style="list-style-type: none"> · 개인 파일 업로드, 다운로드 제공 · 개인 저장 공간을 가지고 있음 	
차이점	<ul style="list-style-type: none"> · 사용자 계정 없이는 파일공유를 지원하지 않음 	<ul style="list-style-type: none"> · SNS를 통한 사용자간의 파일 공유

2.2 클라우드 컴퓨팅

클라우드 컴퓨팅(Cloud Computing)이란 각 PC 단말에서 개별적으로 프로그램을 설치해 데이터를 저장하던 기존 방식에서 벗어나, 인터넷 네트워크상에 모든 컴퓨팅 자원을 저장하여 개별 컴퓨터에 할당하는 개념이다. (그림 1)은 클라우드 컴퓨팅 개념을 잘 보여주고 있다. 본 시스템은 이 기술을 이용하여 일정량의 컴퓨팅 자원을 사용자에게 할당하고 데이터 스토리지 및 사용자간 공유 서비스를 제공한다[3][4].



(그림 1) 클라우드 컴퓨팅의 개념

3. 세부 설계 및 구현

3.1 본 App 에서의 클라우드 시스템의 특징

해당 시스템의 클라우드 기능으로 내세운 것은 웹하드 기능이다. 기존의 스마트폰을 이용한 웹하드는 사용자 개인의 파일만 다운로드 업로드가 가능하고 파일의 공유를 할 수 없었다. 또한 기존 SNS에서도 사진과 동영상 보기만을 지원한다. 해당 시스템은 SNS를 이용하여 사진, 동영상 보기를 지원할 뿐만 아니라 개인 파일을 공유할 수 있음과 동시에 파일에 대한 설명을 첨부하거나 의견을 제시할 수도 있다.

3.2 개발환경

<표4>은 본 논문의 App 개발을 위한 개발환경이다

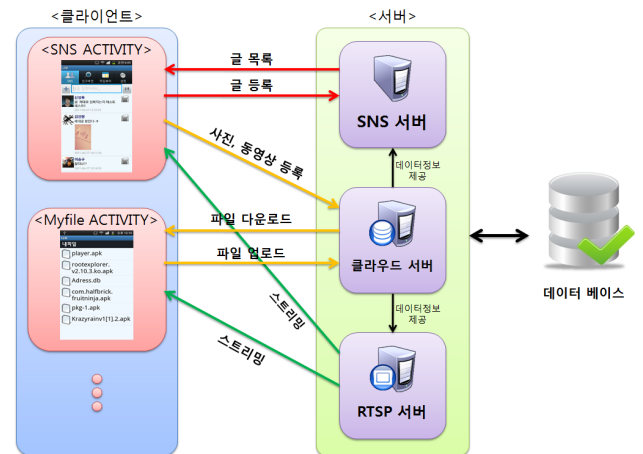
<표4> 본 논문의 App 개발환경

서버		클라이언트	
OS	Windows 7.	OS	Windows 7.
개발 도구	Microsoft Visual Studio 2010. MySQL API. RTSP 모듈.	개발 도구	Eclipse Indigo. SQLite.
개발 언어	C언어	개발 언어	Java.

DBMS	
운영체제	Windows 7.
개발도구	MySQL Community Server 5.5.14.
개발언어	SQL.

3.3 시스템 구성

시스템 구성은 SNS서버, 클라우드 서버, 스트리밍 서비스를 제공하는 RTSP 서버와 DBMS로 분류된다. (그림 2)은 본 논문의 App 시스템 구성도이다.



(그림 2) 시스템 구성도

3.4 주요 구현 사항

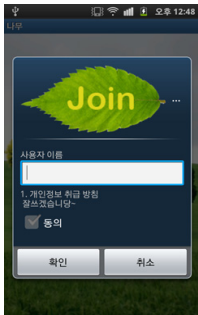
3.4.1 로그인(Login)

로그인 액티비티에서는 회원가입 기능을 제공한다. 회원가입이 되어있을 경우 바로 로그인을 수행한다. (그림 3)와 같이 단말기의 전화번호를 얻어오고 로그인 요청과 함께 사용자를 식별하기위해 서버로 전송한다.

```
// 공유변수
Condination cd = new Condination();
// 휴대폰 정보 가져오기
TelephonyManager mTelephonyMgr = (TelephonyManager) getSystemService(
    Context.TELEPHONY_SERVICE);
cd.imei = mTelephonyMgr.getSubscriberId();
cd.phoneNumber = mTelephonyMgr.getLine1Number();
```

(그림 3) 전화번호 가져오기 소스

(그림 4)에서는 회원가입이 되어있지 않은 최초 접속 시 회원가입 다이얼로그를 띄워 사용자의 이름을 입력받고 정보수집 사용에 동의를 하면 로그인을 한다.



(그림 4) 회원가입 화면

로그인을 하는 동시에 (그림 5)처럼 사용자 단말기에 저장되어있는 전화번호 목록을 읽어와 자체적으로 DB를 관리하며 서버에 자동으로 등록을 한다. 서버에서는 따로 DB를 생성하고 전화번호 목록을 관리한다.

```
// 해당 Contact에 대한 전화번호가 저장되어 있으면 전화번호 목록 출력
if (Integer.parseInt(hasPhone) == 1) {
    Cursor phoneCursor = getContentResolver()
        .query(ContactsContract.CommonDataKinds.Phone.CONTENT_URI, null,
            ContactsContract.CommonDataKinds.Phone.CONTACT_ID+"=?", ContactID, null, null);
    while (phoneCursor.moveToNext()) {
        String number = phoneCursor.getString(phoneCursor
            .getColumnIndex(ContactsContract.CommonDataKinds.Phone.NUMBER));
        String numberType = phoneCursor.getString(phoneCursor
            .getColumnIndex(ContactsContract.CommonDataKinds.Phone.TYPE));
        switch (Integer.parseInt(numberType)) {
            default:
                db = mHelper.getReadableDatabase();
                Cursor cursor;
                cursor = db.rawQuery("SELECT pnumber FROM adr", null);
                boolean Result = false;
                while (cursor.moveToNext()) {
                    String pnumber = cursor.getString(0);
                    if (pnumber.equals(number) == true) {
                        Result = true;
                        break;
                    }
                }
                cursor.close();
                if (Result == false) { // 전화번호부에 하나도 일치하는게 없으면 서버에 추천목록에 추가.
                    if (socket.isConnected())
                        socket.send("1003" + (String) cd.imei+ number + name + "");
                    // DB 삽입
                    db = mHelper.getWritableDatabase();
                    String temp = "INSERT INTO adr VALUES (null, " + number + ")";
                    db.execSQL(temp);
                    Result = false;
                }
                break;
            }
        }
    mHelper.close();
    phoneCursor.close();
}
```

(그림 5) 전화번호부 등록 소스

3.4.2 SNS

(그림 6)에서와 같이 SNS 액티비티는 자기 자신이나 친구로 추가된 사용자가 올린 글이나 사진 동영상등을 볼 수 있는 기능을 제공한다. 작성된 글의 댓글 항목을 터치하면 해당 글에 댓글을 입력하거나 볼 수 있으며 디스켓 모양 이미지를 터치하면 해당 사용자가 업로드 한 파일을 다운 받을 수 있다.



(그림 6) SNS 액티비티 화면

(그림 7)은 BitmapDownloaderTask 클래스로서 이미지를 비동기적으로 받아오는 클래스이다. SNS 액티비티 왼쪽에 사용자 사진이나 사용자가 업로드한 사진을 불러올 때 비트맵으로 디코딩 하기 때문에 리스트뷰의 속도 저하를 발생시킨다. 이를 방지하여 사용자에게 편의성을 제공하기 위해서 데이터를 받아올 때 통신망의 부하가 적은 텍스트는 먼저 받아오고, AsyncTask 이용으로 이미지를 비동기적으로 받아온다.

```
class BitmapDownloaderTask extends AsyncTask<String, Void, Bitmap> {
    private String url;
    private final WeakReference<ImageView> imageViewReference;

    public BitmapDownloaderTask(ImageView imageView) {
        imageViewReference = new WeakReference<ImageView>(imageView);
    }

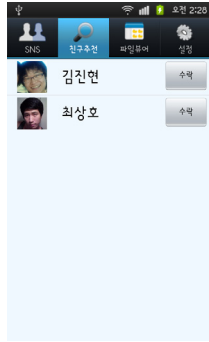
    @Override
    protected Bitmap doInBackground(String... params) {
        url = params[0];
        return downloadBitmap(url);
    }

    @Override
    protected void onPostExecute(Bitmap bitmap) {
        if (isCancelled()) {
            bitmap = null;
        }
        addBitmapToCache(url, bitmap);
        if (imageViewReference != null) {
            ImageView imageView = imageViewReference.get();
            BitmapDownloaderTask bitmapDownloaderTask
                = getBitmapDownloaderTask(imageView);
            if (this == bitmapDownloaderTask) {
                imageView.setImageBitmap(bitmap);
            }
        }
    }
}
```

(그림 7) BitmapDownloaderTask 클래스 소스

3.4.3 친구추천

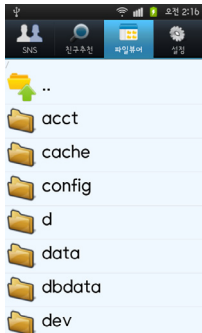
사용자 단말기의 주소록에 있는 사람이 본 어플리케이션 사용을 위해 가입을 할 경우 추천 친구 목록에 자동으로 추가된다. (그림 8)에서 수락 버튼을 누르면 친구추가가 이루어지고 친구 목록에 추가되며 SNS 글 목록에 추가한 친구가 작성한 글까지 출력이 된다.



(그림 8) 친구추천 액티비티 화면

3.4.4 파일뷰어 및 내 파일

(그림 9, 10)에서와 같이 파일뷰어는 단말기 내의 파일 목록을 보여주고 파일을 내 파일에 업로드 할 수 있는 기능 가지고 있다. 파일 업로드 선택 시 데이터 서버에 파일을 업로드하고 내 파일 목록에 보여진다.



(그림 9) 파일뷰어



(그림 10) 내파일 화면

동영상의 경우 RTSP 서버를 통해 스트리밍 서비스를 지원받고 내 파일의 동영상을 실행할 시 (그림 11)에서처럼 동영상을 재생한다.

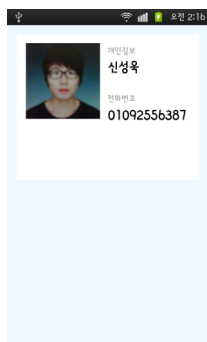
```
//동영상 재생
Intent intent = new Intent(Intent.ACTION_VIEW);
Uri uri = Uri.parse(Uri.encode(rtsp_url));
intent.setDataAndType(uri, "video/*");
startActivity(intent);
```

(그림 11) 동영상 재생 소스

3.4.5 설정



(그림 12) 설정 화면



(그림 13) 내 프로필 화면

(그림 12)설정에는 공지사항, 버전정보, 내 프로필 등록 기능이 있다. (그림 13)에서와 같이 사용자 프로필 사진을 설정 할 수 있으며 프로필 사진을 터치 할 시에는 프로필 사진을 변경할 수 있다.

(그림 14)는 프로필 사진 변경 기능을 제공한다. 사진앨범 선택 시에는 안드로이드에서 제공하는 갤러리에서 사진을 가져올 수 있으며 카메라 선택 시에는 카메라를 실행하여 즉석으로 촬영한 사진을 업로드할 수 있다.

```
public void onClick(View arg0) {
    new AlertDialog.Builder(MyProfile.this).setTitle("사진선택")
        .setItems(R.array.Profile_img_select,
            new DialogInterface.OnClickListener() {
                @Override
                public void onClick(DialogInterface dialog,
                    int which) {
                    if (which == 0) // 사진앨범 선택시
                    {
                        Intent intent = new Intent();
                        intent.setAction(Intent.ACTION_GET_CONTENT);
                        intent.setType("image/*");
                        startActivityResult(intent, REQ_GALLERY);
                    }
                    else if (which == 1) // 카메라 선택시
                    {
                        Intent intent;
                        intent = new Intent(MyProfile.this, SHCamera.class);
                        startActivity(intent);
                    }
                }
            })
        .show();
}
```

(그림 14) 프로필 사진 변경 다이얼로그 소스

4. 결론 및 향후 연구과제

이 연구를 통해 개발된 App을 이용하게 되면 클라우드 서비스를 제공받고 소셜 네트워크에 적극 활용 할 수 있으므로 언제 어디서나 사용자 간에 정보를 공유함과 동시에 데이터와 미디어를 공유할 수 있어 강력한 인간관계를 형성함과 동시에 손쉬운 파일 공유와 편리함을 제공한다. 스마트폰을 통한 실시간 파일 공유가 가능하기 때문에 활용도는 무궁무진하며 여러 용도로 사용이 가능하다. 커뮤니티 적인 면에서도 활용도가 높다.

현재 개발된 App은 사용자들 간의 파일의 공유에 초점이 맞추어져 구현되었다. 사용자가 공유를 원치 않는 파일은 공유를 하지 않게 공유설정과 권한 설정 기능을 더 제공하는 등 개선해야할 필요성이 있으며 더 편리한 사용 환경을 위해 깔끔한 사용자 인터페이스를 구성할 필요가 있다. 또한 '페이스북'의 친구 추천기능처럼 더욱 강력한 친구추천 기능을 제공하여 SNS측면에서 서비스를 개선할 필요가 있다.

참고문헌

[1] 스마트폰 시장 증가율 “<http://samsungamoled.net/259>”.
 [2] OS별 사용 현황 “<http://pch1286.blog.me/80131080987>”.
 [3] 한국콘텐츠진흥원, “클라우드 컴퓨팅 기술 동향”, 문화기술 심층리포트 11호, 2011.2.
 [4] 스마트폰 성장의 촉매제 - 클라우드 컴퓨팅 “<http://gojump0713.blog.me/140112653566>”.