

모바일 프로그래밍 (나의 앱 만들기)

2017.6.19

학과	컴퓨터공학과
학번	201333526
성명	김형준



목 차

1. 나의 앱 만들기.....	3
1.1 애플리케이션 개요	3
1.2 애플리케이션 구성	4
1.3 애플리케이션 실행화면	6
1.4 데이터베이스 설계	10
1.5 애플리케이션 구현	12
2. 나의 앱 만들기 후기.....	93

1. 나의 앱 만들기

1.1 애플리케이션 개요

■ 어플리케이션 명

『 Place 』

■ 애플리케이션 요약

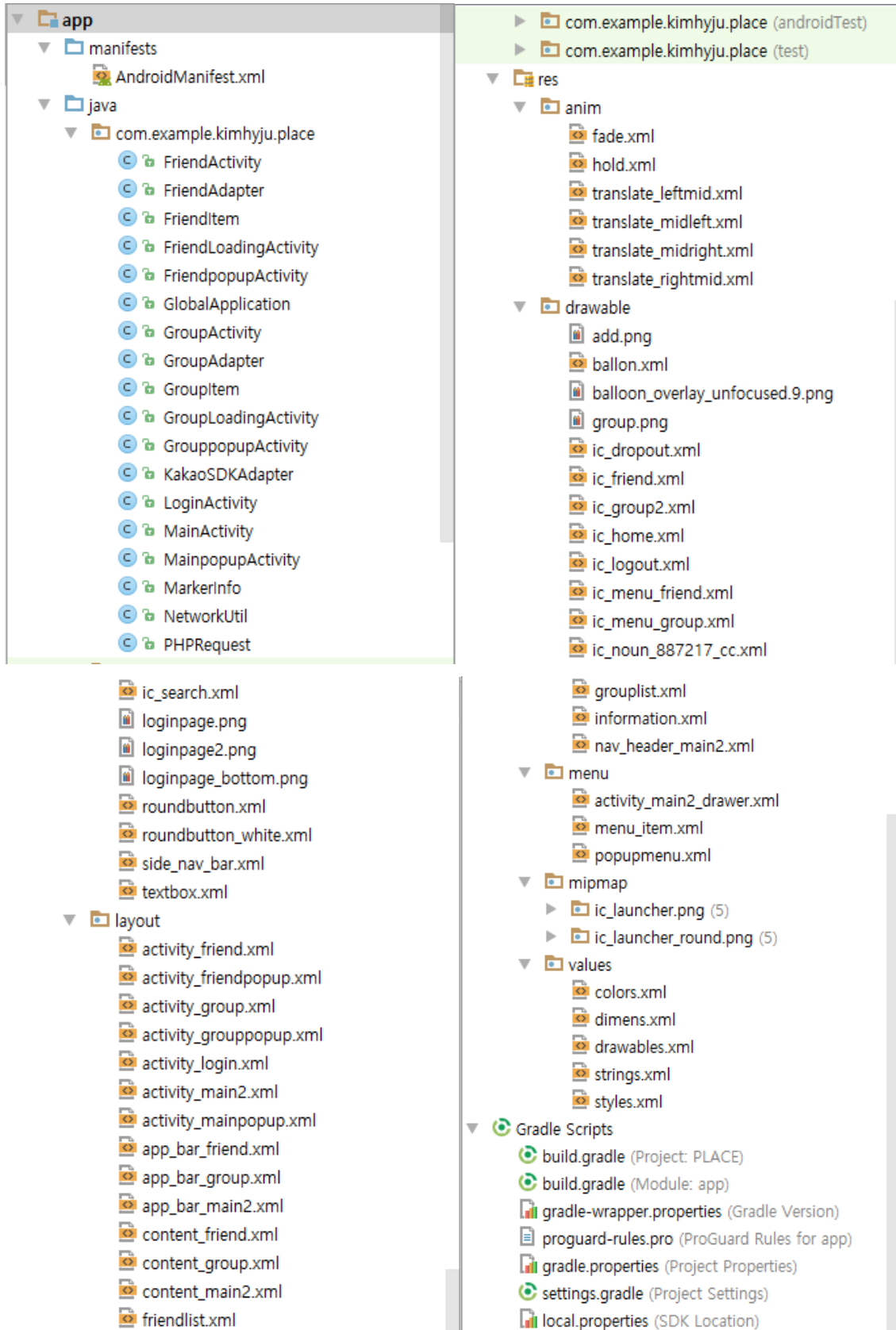
저희가 준비한 어플리케이션 이름은 ‘Place’입니다. 이 앱은 Google maps API를 활용하여서 지도 위에 마커를 찍고, 그 마커들을 친구들과 그룹을 지어 공유하는 어플리케이션 입니다.

‘Place’앱은 Google maps를 이용하여 자신의 위치 뿐만 아니라 위치 검색기능과 마커를 찍고 저장해 놓는 기능을 가지고 있습니다. 또 친구목록에 친구들과 그룹을 만들어서 마커를 공유하여 네트워킹 기능을 할 수 있도록 합니다.

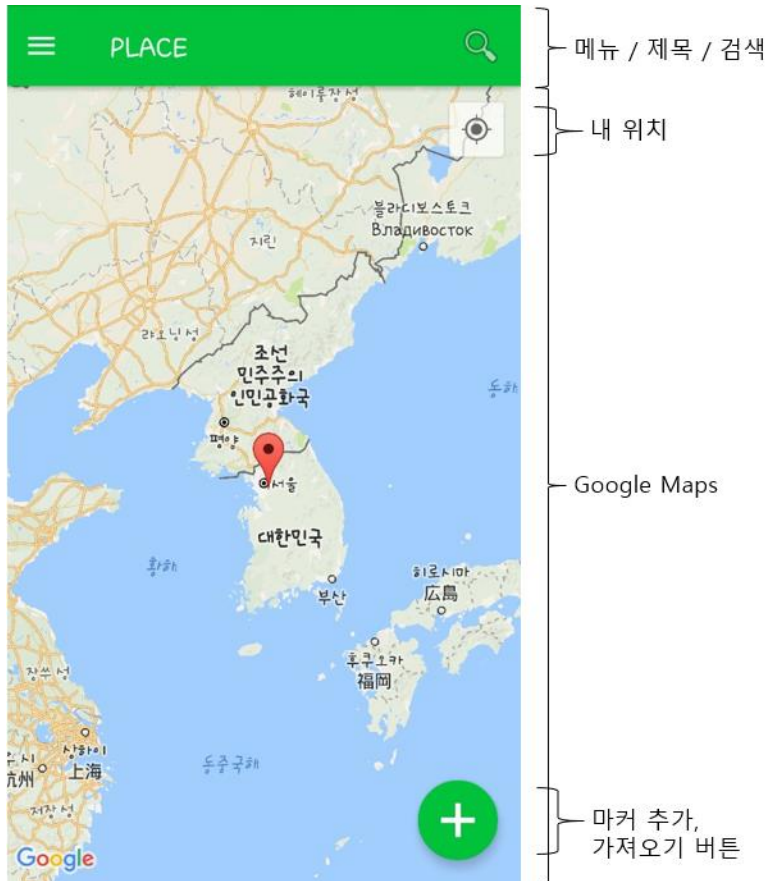
1. 지도 검색 기능 : 구글 Maps api를 이용하여 구글 지도 내 검색 및 현재 자신의 위치를 알 수 있습니다.
2. 지도 내 마커 표시 : 지도에 원하는 곳에 마커를 새겨 나만의 지도를 완성할 수 있습니다. 혹은 그룹의 지도를 만들 수 있습니다.
3. 네트워킹 기능 : 지도 내 마커를 표시하여 사용자끼리 위치정보를 공유하며 네트워킹 활동을 할 수 있습니다.

1.2 애플리케이션 구성

■ 프로젝트 구성도



■ 메인 화면 구성도







맨 위의 메뉴 아이콘을 누르면 사용자 프로필, 메뉴(홈, 친구목록, 그룹목록, 로그아웃, 탈퇴하기)가 나오며 검색 아이콘을 누르면 지역을 검색할 수 있다.

내 위치는 현재 내 위치를 GPS를 이용하여 구글 지도에 찍어주고 우측하단의 버튼을 통하여 지도에 마커를 새롭게 추가하거나 다른 그룹에서 추가한 마커를 가져올 수 있다.

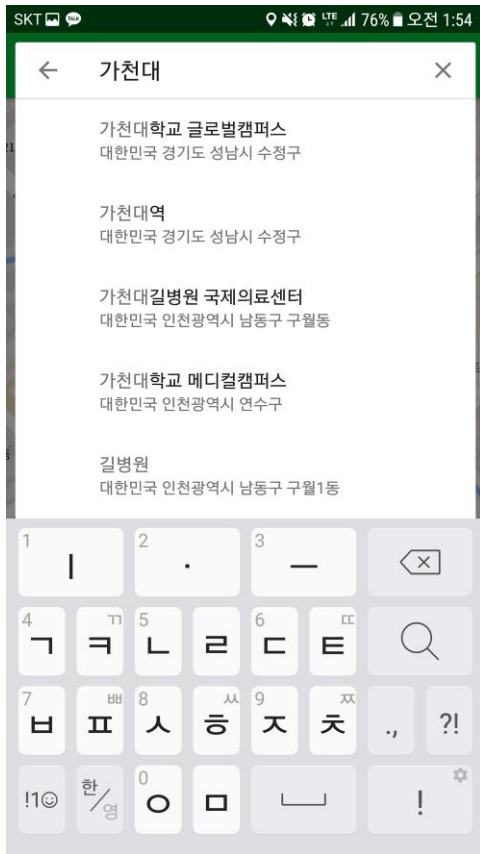
■ 실행 기능

주요기능	상세기능	비고
GPS로 위치확인	Google 지도안에서 자신의 위치 표시	메인 화면
친구 목록 확인	자신이 추가한 친구 목록을 확인	친구 목록
그룹 목록 확인	자신이 가입한 그룹 목록 확인	그룹 목록
마커 추가	지도에 원하는 내용으로 마커 추가하기	지도 내 추가
마커 가져오기	자신이 가입한 그룹의 마커 가져오기	지도 동기화
위치 검색	구글 지도에서 위치 검색 기능	메인 화면
그룹 생성하기	새로운 그룹 생성하기	그룹 기능
그룹 추가하기	그룹 가입하기	그룹 기능
친구 추가	ID입력하여 친구추가하기	친구 기능

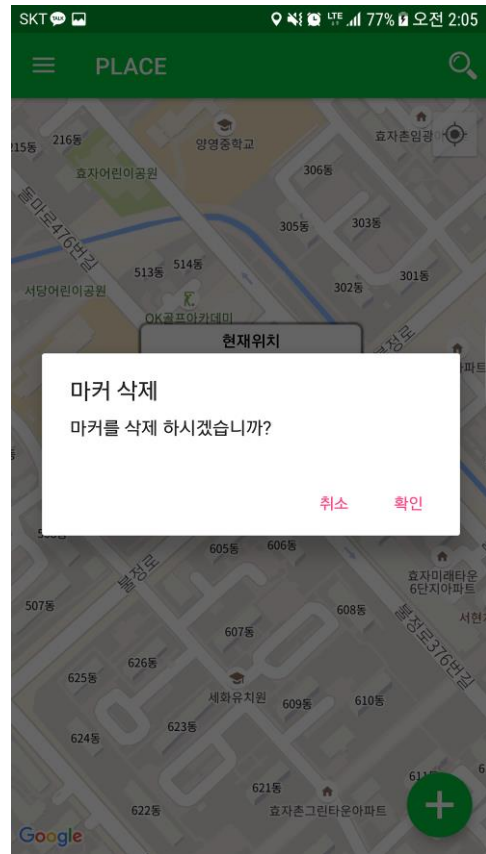
1.3 애플리케이션 실행화면

화면1	화면2
	
로그인 및 회원가입 화면	메인화면(구글 지도)
화면3	화면4
	
메인화면(GPS버튼 클릭)	메인화면(마커 불러온 후)

화면5



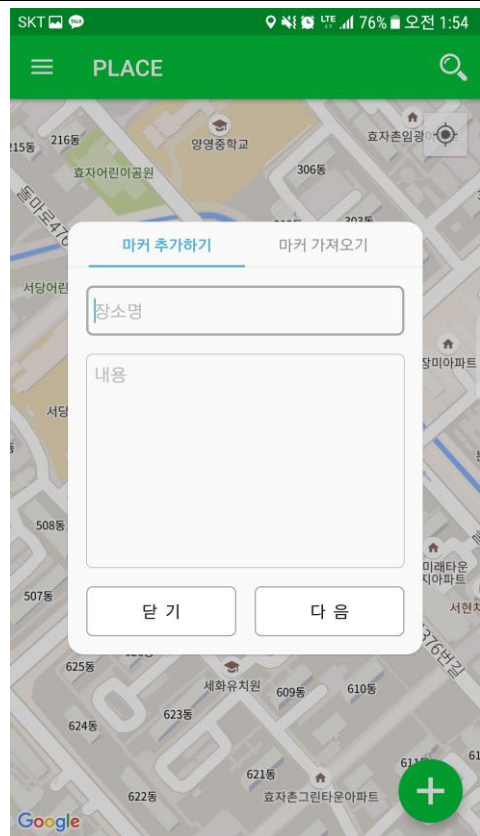
화면6



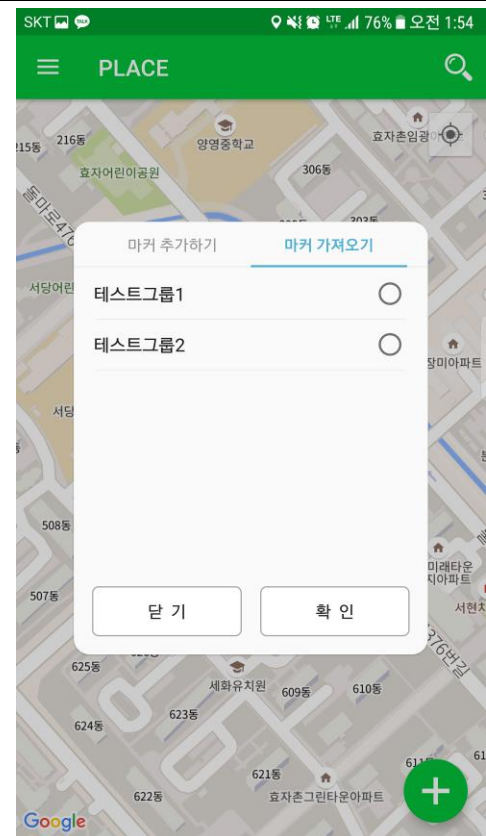
메인화면(장소검색)

메인화면(마커 삭제)

화면7

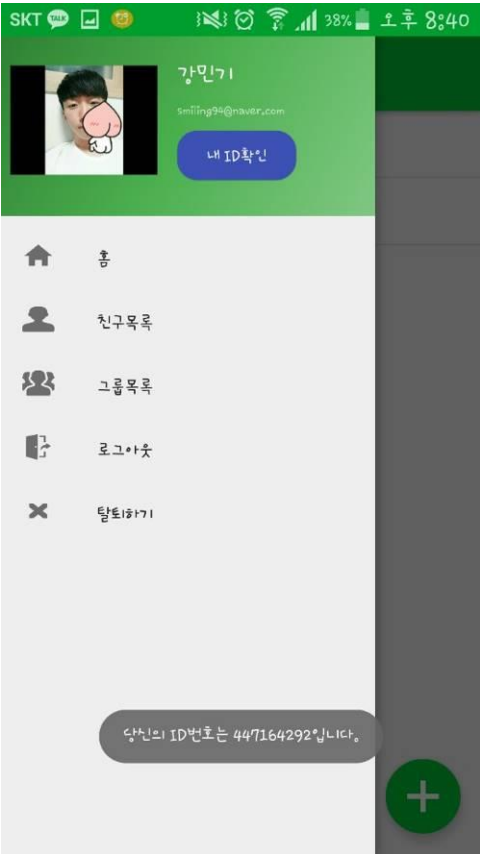
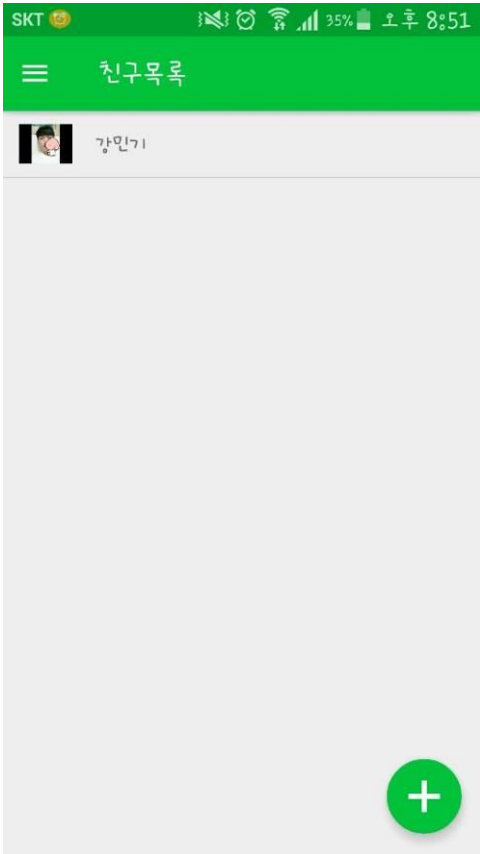

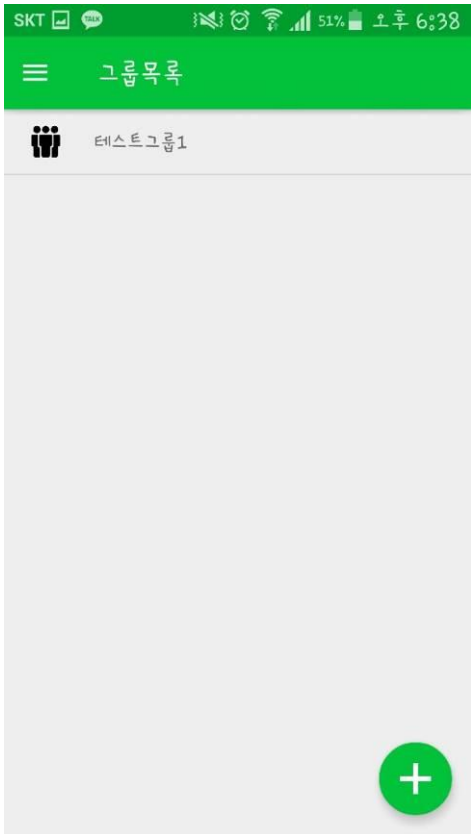


화면8



메인화면(버튼 기능1)

메인화면(버튼 기능2)

화면9	화면10
	
네비게이션 바(내 ID확인 버튼 누른 후)	친구목록 화면
화면11	화면12
	
친구목록 친구추가 버튼	그룹목록 화면

화면13	화면14
	
그룹목록(버튼 기능1)	그룹목록(버튼 기능2)

1.4 데이터베이스 설계

■ 데이터베이스 명 : appdata.db

□ 테이블명 : user

Name	Type	Key	Etc
Userid	Int(10)	PK	NOT NULL
Imgurl	Text		Default NULL
Name	Text		NOT NULL

□ 테이블명 : group_list

Name	Type	Key	Etc
Group_name	Text	PK	NOT NULL
Group_password	Text		NOT NULL

□ 테이블명 : group_relation

Name	Type	Key	Etc
Relationid	Text	PK	NOT NULL
Group_name	Text		NOT NULL
Userid	Int(10)		NOT NULL

□ 테이블명 : friend_realtion

Name	Type	Key	Etc
Friendid	double	PK	NOT NULL
Me_name	Varchar(20)		NOT NULL
Me_userid	Int(10)		NOT NULL
Friend_name	Varchar(20)		NOT NULL
Friend_userid	Int(10)		NOT NULL
Friend_image	Text		NOT NULL


□ 테이블명 : marker

Name	Type	Key	Etc
Markerid	Varchar(40)	PK	NOT NULL
Lat	Double		Default NULL
Lng	Double		Default NULL
Name_place	Varchar(20)		NOT NULL
Content	Text		NOT NULL
Name_group	Varchar(20)		Default NULL
Name_user	Varchar(20)		Default NULL
Userid	Int(10)		Default NULL

1.5 애플리케이션 구현

1) 로그인화면

■ 레이아웃 구현

화면(UI)	설명
	<ul style="list-style-type: none"> - 앱을 실행시키면처음 나타나는 화면 - 앱의 로고 이미지로 바탕화면을 구성 - 아래쪽에 카카오톡으로 로그인하는 버튼 구현
activity_login.xml	
<pre> <?xml version="1.0" encoding="utf-8" ?> <android.support.constraint.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android" xmlns:app="http://schemas.android.com/apk/res-auto" xmlns:tools="http://schemas.android.com/tools" android:layout_width="match_parent" android:layout_height="match_parent" android:background="@drawable/loginpage" tools:context="com.example.kimhyju.place.LoginActivity"> <com.kakao.usermgmt.LoginButton android:id="@+id/com_kakao_login" android:layout_width="320dp" android:layout_height="wrap_content" android:layout_gravity="bottom" android:layout_marginBottom="36dp" android:layout_marginLeft="8dp" android:layout_marginRight="8dp" android:visibility="invisible" app:layout_constraintBottom_toBottomOf="parent" app:layout_constraintHorizontal_bias="0.493" app:layout_constraintLeft_toLeftOf="parent" app:layout_constraintRight_toRightOf="parent" /> </android.support.constraint.ConstraintLayout> </pre>	

■ 기능 구현

LoginActivity.java

```

package com.example.kimhyju.place;

import android.Manifest;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.os.Bundle;
import android.os.Handler;
import android.support.v4.app.ActivityCompat;
import android.support.v4.content.ContextCompat;
import android.support.v7.app.AppCompatActivity;
import android.util.Log;
import android.view.View;
import android.view.Window;
import android.view.WindowManager;
import android.widget.Toast;

import com.kakao.auth.ErrorCode;
import com.kakao.auth.ISessionCallback;
import com.kakao.auth.Session;
import com.kakao.network.ErrorResult;
import com.kakao.usermgmt.UserManagement;
import com.kakao.usermgmt.callback.MeResponseCallback;
import com.kakao.usermgmt.response.model.UserProfile;
import com.kakao.util.exception.KakaoException;
import com.kakao.util.helper.log.Logger;

import java.net.MalformedURLException;

public class LoginActivity extends AppCompatActivity {
    SessionCallback callback;
    com.kakao.usermgmt.LoginButton loginButton;
    long userID = 0; //사용자 고유번호
    String pImage = ""; //사용자 프로필 경로
    String nickName;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        supportRequestWindowFeature(Window.FEATURE_NO_TITLE);
        NetworkUtil.setNetworkPolicy();
        getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN,
            WindowManager.LayoutParams.FLAG_FULLSCREEN);
        setContentView(R.layout.activity_login);
        loginButton=(com.kakao.usermgmt.LoginButton)findViewById(R.id.com_kakao_login);
        loginButton.setVisibility(View.INVISIBLE);
        Handler handler=new Handler();
        checkDangerousPermissions();
        Session.getCurrentSession().checkAccessTokenInfo();

        //이미 로그인되어있는지를 판단, 로그인화면을 1초동안 보여줌
        if(Session.getCurrentSession().isOpened()==true){
            handler.postDelayed(new Runnable() {
                @Override
                public void run() {
                    Intent intent = new Intent(LoginActivity.this, MainActivity.class);
                    startActivity(intent);
                    overridePendingTransition(R.anim.fade,R.anim.hold);
                    finish();
                }
            }, 1000);
        }
    }
}

```

```

    }
    }, 1000);

    }else{
        loginButton.setVisibility(View.VISIBLE);
    }

    callback = new SessionCallback();
    Session.getCurrentSession().addCallback(callback);
}

//재 로그인 요청
private void redirectLoginActivity() {
    final Intent intent = new Intent(this, LoginActivity.class);
    startActivity(intent);
    overridePendingTransition(R.anim.fade, R.anim.hold);
    finish();
}

//간편로그인시 호출되는 부분
@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    //간편로그인시 호출, 없으면 간편로그인시 로그인 성공화면으로 넘어가지 않음
    if (Session.getCurrentSession().handleActivityResult(requestCode, resultCode, data)) {
        return;
    }

    super.onActivityResult(requestCode, resultCode, data);
}

@Override
protected void onDestroy() {
    super.onDestroy();
    Session.getCurrentSession().removeCallback(callback);
}

//SessionCallback 클래스 구현
private class SessionCallback implements ISessionCallback {

    @Override
    public void onSessionOpened() {

        UserManagement.requestMe(new MeResponseCallback() {

            @Override
            public void onFailure(ErrorResult errorResult) {
                String message = "failed to get user info. msg=" + errorResult;
                Logger.d(message);

                ErrorCode result = ErrorCode.valueOf(errorResult.getErrorCode());
                if (result == ErrorCode.CLIENT_ERROR_CODE) {
                    finish();
                } else {
                    //재 로그인
                    Toast.makeText(getApplicationContext(), "다시 로그인
해주세요.", Toast.LENGTH_SHORT).show();
                    redirectLoginActivity();
                }
            }

            @Override
            public void onSessionClosed(ErrorResult errorResult) {

```

```

    }

    @Override
    public void onNotSignedUp() {
    }

    @Override
    public void onSuccess(UserProfile userProfile) {
        //로그인 성공 시 로그인한 사용자의 일련번호, 닉네임, 이미지 url 리턴
        //사용자 캐시 정보 업데이트
        if (userProfile != null) {
            userProfile.saveUserToCache();
        }
        Logger.e("succeeded to update user profile",userProfile,"Wn");
        ///////////////

        nickname = userProfile.getNickname();//닉네임
        userID = userProfile.getId();//사용자 고유번호
        plmage = userProfile.getProfileImagePath();//사용자 프로필 경로*/
        Log.e("UserProfile", userProfile.toString());//전체 정보 출력
        insertuser();
        Intent intent = new Intent(LoginActivity.this, MainActivity.class);
        startActivity(intent);
        overridePendingTransition(R.anim.fade,R.anim.hold);
        finish();
    }
}

@Override
public void onSessionOpenFailed(KakaoException exception) {
    // 세션 연결이 실패했을때
    if(exception != null) {
        Toast.makeText(getApplicationContext(),"세션 연결 실패",Toast.LENGTH_SHORT).show();
        Logger.e(exception);
    }
}

//카카오톡으로 로그인하면 데이터베이스에 사용자 정보를 저장
public void insertuser(){
    try{
        PHPRequest request=new PHPRequest("http://180.71.13.212:8181/insert.php");
        String
        result=request.PhPinsert(String.valueOf(userID),String.valueOf(plmage),String.valueOf(nickname));
        if(result.equals("1")){
            Toast.makeText(getApplicationContext(),"가입되었습니다!",Toast.LENGTH_LONG).show();
        }else if(result.equals("-1")){
        }
    }catch (MalformedURLException e){
        e.printStackTrace();
    }
}

//권한 체크
private void checkDangerousPermissions() {
    String[] permissions = {//import android.Manifest;
        Manifest.permission.ACCESS_FINE_LOCATION,
        Manifest.permission.ACCESS_COARSE_LOCATION,
        Manifest.permission.READ_EXTERNAL_STORAGE,
        Manifest.permission.WRITE_EXTERNAL_STORAGE,
    }
}

```



```

        Manifest.permission. INTERNET//쓰기 권한
    };

    int permissionCheck = PackageManager.PERMISSION_GRANTED;
    for (int i = 0; i < permissions.length; i++) {
        permissionCheck = ContextCompat.checkSelfPermission(this, permissions[i]);
        if (permissionCheck == PackageManager.PERMISSION_DENIED) {
            break;
        }
    }

    if (permissionCheck == PackageManager.PERMISSION_GRANTED) {
    } else {
        if (ActivityCompat.shouldShowRequestPermissionRationale(this, permissions[0])) {
        } else {
            ActivityCompat.requestPermissions(this, permissions, 1);
        }
    }
}

// 사용자의 권한 확인 후 사용자의 권한에 대한 응답 결과를 확인하는 콜백 메소드
@Override
public void onRequestPermissionsResult(int requestCode, String[] permissions,
                                       int[] grantResults) {
    if (requestCode == 1) {
        for (int i = 0; i < permissions.length; i++) {
            if (grantResults[i] == PackageManager.PERMISSION_GRANTED) {
            } else {
            }
        }
    }
}
}

```

2) 카카오톡 API

■ 기능 구현

GlobalApplication.java

```

package com.example.kimhyju.place;

import android.app.Activity;
import android.app.Application;

import com.kakao.auth.KakaoSDK;

public class GlobalApplication extends Application {

    private static volatile GlobalApplication instance = null;
    private static volatile Activity currentActivity = null;

    @Override
    public void onCreate() {

        super.onCreate();
        instance = this;
        KakaoSDK.init(new KakaoSDKAdapter());
    }

    public static GlobalApplication getGlobalApplicationContext(){
        return instance;
    }
}

```

```

    }

    public static void setCurrentActivity(Activity currentActivity){
        GlobalApplication.currentActivity = currentActivity;
    }

    public static Activity getCurrentActivity(){
        return currentActivity;
    }
}

```

KaKaoSDKAdapter.java

```

package com.example.kimhyju.place;

import android.app.Activity;
import android.content.Context;
import com.kakao.auth.ApprovalType;
import com.kakao.auth.AuthType;
import com.kakao.auth.IApplicationConfig;
import com.kakao.auth.ISessionConfig;
import com.kakao.auth.KakaoAdapter;

public class KakaoSDKAdapter extends KakaoAdapter {
    /**
     * Session Config 에 대해서는 default 값들이 존재한다.
     * 필요한 상황에서만 override 해서 사용하면 됨.
     * @return Session 의 설정값.
     */
    @Override
    public ISessionConfig getSessionConfig() {
        return new ISessionConfig() {
            @Override
            public AuthType[] getAuthTypes() {
                return new AuthType[] {AuthType.KAKAO_TALK};
            }

            @Override
            public boolean isUsingWebViewTimer() {
                return false;
            }

            @Override
            public boolean isSecureMode() {
                return false;
            }

            @Override
            public ApprovalType getApprovalType() {
                return ApprovalType.INDIVIDUAL;
            }

            @Override
            public boolean isSaveFormData() {
                return true;
            }
        };
    }

    @Override
    public IApplicationConfig getApplicationConfig() {
        return new IApplicationConfig() {
            public Activity getTopActivity() {
                return GlobalApplication.getCurrentActivity();
            }
        };
    }
}

```

```

    }

    @Override
    public Context getApplicationContext() {
        return GlobalApplication.getGlobalApplicationContext();
    }
};
}
}

```

3) 메인화면

■ 레이아웃 구현

화면(UI)	설명
	<ul style="list-style-type: none"> - 메인화면은 메뉴버튼, FloatingActionButton, 검색버튼, 앱 제목, 구글지도로 이루어져있다. - 플로팅버튼을 클릭하여 마커를 가져오면 오른쪽 화면처럼 마커가 지도에 표시되고 클릭하면 정보가 뜬다. - 검색버튼을 누르면 구글API를 이용하여 웹상의 구글지도에서 검색하는 것과 똑 같은 검색결과들이 나타난다. - 메뉴버튼을 누르면 친구목록, 그룹목록, 홈, 로그아웃, 탈퇴하기 메뉴가 있다.

activity_main2.xml

```

<?xml version="1.0" encoding="utf-8" ?>
<android.support.v4.widget.DrawerLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/drawer_layout"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:fitsSystemWindows="true"
    tools:openDrawer="start">

    <include
        layout="@layout/app_bar_main2"
        android:layout_width="match_parent"
        android:layout_height="match_parent" />

    <android.support.design.widget.NavigationView
        android:id="@+id/nav_view"
        android:layout_width="wrap_content"
        android:layout_height="match_parent"
        android:layout_gravity="start"
        android:fitsSystemWindows="true"
        app:headerLayout="@layout/nav_header_main2"
        app:menu="@menu/activity_main2_drawer" />

</android.support.v4.widget.DrawerLayout>

```

app_bar_main2.xml

```

<?xml version="1.0" encoding="utf-8"?>
<android.support.design.widget.CoordinatorLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="com.example.kimhyju.place.MainActivity">

    <android.support.design.widget.AppBarLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:theme="@style/AppTheme.AppBarOverlay">

        <android.support.v7.widget.Toolbar
            android:id="@+id/toolbar1"
            android:layout_width="match_parent"
            android:layout_height="?attr/actionBarSize"
            android:background="?attr/colorPrimary"
            app:popupTheme="@style/AppTheme.PopupOverlay" />

    </android.support.design.widget.AppBarLayout>

    <include layout="@layout/content_main2"
        android:id="@+id/include" />

    <android.support.design.widget.FloatingActionButton
        android:id="@+id/fab"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="bottom|end"
        android:layout_margin="@dimen/fab_margin"
        app:backgroundTint="?attr/colorPrimary"
        app:srcCompat="@drawable/add" />

</android.support.design.widget.CoordinatorLayout>

```

content_main2.xml

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    app:layout_behavior="@string/appbar_scrolling_view_behavior"
    tools:context="com.example.kimhyju.place.MainActivity">

    <fragment
        android:id="@+id/map"
        android:name="com.google.android.gms.maps.SupportMapFragment"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_weight="1" />

</LinearLayout>

```

■ 기능 구현

MainActivity.java

```
package com.example.kimhyju.place;

import android.Manifest;
import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.location.Geocoder;
import android.location.Location;
import android.location.LocationListener;
import android.location.LocationManager;
import android.os.Bundle;
import android.os.Handler;
import android.support.annotation.NonNull;
import android.support.design.widget.FloatingActionButton;
import android.support.design.widget.NavigationView;
import android.support.v4.content.ContextCompat;
import android.support.v4.view.GravityCompat;
import android.support.v4.widget.DrawerLayout;
import android.support.v7.app.ActionBarDrawerToggle;
import android.support.v7.app.AlertDialog;
import android.support.v7.app.AppCompatActivity;
import android.support.v7.widget.Toolbar;
import android.util.Log;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.TextView;
import android.widget.Toast;

import com.google.android.gms.common.ConnectionResult;
import com.google.android.gms.common.GooglePlayServicesNotAvailableException;
import com.google.android.gms.common.GooglePlayServicesRepairableException;
import com.google.android.gms.common.api.GoogleApiClient;
import com.google.android.gms.location.places.Place;
import com.google.android.gms.location.places.Places;
import com.google.android.gms.location.places.ui.PlaceAutocomplete;
import com.google.android.gms.maps.CameraUpdateFactory;
import com.google.android.gms.maps.GoogleMap;
import com.google.android.gms.maps.MapInitializer;
import com.google.android.gms.maps.OnMapReadyCallback;
import com.google.android.gms.maps.SupportMapFragment;
import com.google.android.gms.maps.model.BitmapDescriptorFactory;
import com.google.android.gms.maps.model.LatLng;
import com.google.android.gms.maps.model.Marker;
import com.google.android.gms.maps.model.MarkerOptions;
import com.kakao.auth.Session;
import com.kakao.network.ErrorResult;
import com.kakao.usermgmt.UserManagement;
import com.kakao.usermgmt.callback.LogoutResponseCallback;
import com.kakao.usermgmt.callback.MeResponseCallback;
import com.kakao.usermgmt.callback.UnLinkResponseCallback;
import com.kakao.usermgmt.response.model.UserProfile;
import com.kakao.util.helper.log.Logger;
```

```

import java.io.InputStream;
import java.net.MalformedURLException;
import java.net.URL;
import java.util.ArrayList;
import java.util.Locale;

import static com.example.kimhyju.place.R.id.group_list;

public class MainActivity extends AppCompatActivity
    implements NavigationView.OnNavigationItemSelectedListener, View.OnClickListener,
    GoogleMap.OnCameraMoveStartedListener, GoogleApiClient.OnConnectionFailedListener, GoogleMap.OnMapClickListener,
    GoogleMap.OnMarkerClickListener, GoogleMap.InfoWindowAdapter, GoogleMap.OnInfoWindowLongClickListener,
    GoogleMap.OnInfoWindowClickListener{
    View nav_head_view, mView;
    Toolbar toolbar;
    private ImageView profileImage;
    private TextView nameText;
    private Button confirmID;

    String nickName = ""; //닉네임
    long userID = 0; //사용자 고유번호
    String pImage = ""; //사용자 프로필 경로
    String user_email;
    String userID;

    Double melatitude;
    Double melongitude;

    LocationManager manager;
    GPSTListener gpsListener;

    GoogleMap map;
    Geocoder gc;
    GoogleApiClient mGoogleApiClient;
    Marker currentmarker;
    ArrayList<Marker Info> marker_list;
    ArrayList<Marker> markers;
    String currentpoint="";
    String mepoint="";
    String name_place,writer_place,content_place;
    Double lat, lng;
    TextView emailText,place_name,place_writer,place_content;

    Boolean flag=false;
    int PLACE_AUTOCOMPLETE_REQUEST_CODE = 1;
    int PLACE_GETMARKER_REQUEST_CODE=2;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main2);
        NetworkUtil.setNetworkPolicy();
        NavigationView navigationView = (NavigationView) findViewById(R.id.nav_view);
        nav_head_view=navigationView.getHeaderView(0);
        profileImage=(ImageView)nav_head_view.findViewById(R.id.profileImage);
        nameText=(TextView)nav_head_view.findViewById(R.id.nameText1);
        emailText=(TextView)nav_head_view.findViewById(R.id.email1);
        confirmID=(Button)nav_head_view.findViewById(R.id.confirmID);
        toolbar = (Toolbar) findViewById(R.id.toolbar1);
        gc=new Geocoder(this, Locale.KOREAN);
        marker_list=new ArrayList<>();
        markers=new ArrayList<>();
        mGoogleApiClient = new GoogleApiClient

```

```

        .Builder(this)
        .addApi(Places.GEO_DATA_API)
        .addApi(Places.PLACE_DETECTION_API)
        .enableAutoManage(this, this)
        .build();

//MainActivity에서 로그인이 되어있는지 한번더 확인
Session.getCurrentSession().checkAccessTokenInfo();
if(Session.getCurrentSession().isOpened()==false)
{
    //세션이 닫힌 경우 다시 초기 화면으로 이동
    Intent intent = new Intent(MainActivity.this, LoginActivity.class);
    startActivity(intent);
    overridePendingTransition(0,0);
    finish();
}
else{
    //세션이 열린 경우 사용자 정보 가져오기
    requestMe();
}

setSupportActionBar(toolbar);
FloatingActionButton fab = (FloatingActionButton) findViewById(R.id.fab);
//FloatingActionButton 클릭이벤트 구현
fab.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        Intent intent=new Intent(getApplicationContext(),MainpopupActivity.class);
        intent.putExtra("userid",userid);
        if(!currentpoint.equals("")) {
            intent.putExtra("currentpoint", currentpoint);
            intent.putExtra("lat", lat);
            intent.putExtra("lng", lng);
        }else{
            intent.putExtra("currentpoint",mepoint);
            intent.putExtra("lat", melatitude);
            intent.putExtra("lng", melongitude);
        }
        intent.putExtra("username",nickname);
        startActivityForResult(intent,PLACE_GETMARKER_REQUEST_CODE);
    }
});

//네비게이션 뷰 구현
DrawerLayout drawer = (DrawerLayout) findViewById(R.id.drawer_layout);
ActionBarDrawerToggle toggle = new ActionBarDrawerToggle(
    this, drawer, toolbar, R.string.navigation_drawer_open, R.string.navigation_drawer_close);
drawer.setDrawerListener(toggle);
toggle.syncState();
navigationView.setNavigationItemSelectedListener(this);

//구글 지도를 프래그먼트에 띄운다.
SupportMapFragment fragment = (SupportMapFragment)
getSupportFragmentManager().findFragmentById(R.id.map);
fragment.getMapAsync(new OnMapReadyCallback() {
    @Override
    public void onMapReady(GoogleMap googleMap) {
        map=googleMap;

        if (ContextCompat.checkSelfPermission(MainActivity.this,
Manifest.permission.ACCESS_FINE_LOCATION)
            == PackageManager.PERMISSION_GRANTED) {
            map.setMyLocationEnabled(true);

```



```

    }
    map.moveCamera(CameraUpdateFactory.newLatLngZoom(new LatLng(37.534892, 126.983210), 5));
//처음시작위치 정의
    map.setOnCameraMoveStartedListener(MainActivity.this); //카메라가 움직이면 실행될 메소드
    map.getUiSettings().setRotateGesturesEnabled(false); //지도가 회전하는기능을 false
    map.setOnMapClickListener(MainActivity.this); //지도를 터치하면 실행되는 동작
    map.setOnMarkerClickListener(MainActivity.this); //마커를 클릭하면 실행되는 동작
    map.setInfoWindowAdapter(MainActivity.this); //마커를 클릭하면 나타나는 정보창을 adapter 를
통해서 구현

    map.setOnInfoWindowLongClickListener(MainActivity.this);
    map.setOnInfoWindowClickListener(MainActivity.this);
}
});

try{
    MapsInitializer.initialize(this);
}catch (Exception e){
    e.printStackTrace();
}
startLocationService();
confirmID.setOnClickListener(this);
}

//네비게이션 뷰에서 내 ID를 확인하는 버튼 구현
@Override
public void onClick(View v) {
    if(v.getId()==R.id.confirmID){
        Toast.makeText(getApplicationContext(), "당신의 ID 번호는
"+user ID+"입니다.", Toast.LENGTH_LONG).show();
    }
}

@SuppressWarnings("StatementWithEmptyBody")
@Override
public boolean onNavigationItemSelected(MenuItem item) {
    // Handle navigation view item clicks here.
    int id = item.getItemId();
    item.setCheckable(false);
    if (id == R.id.friend_list) {
        Intent intent=new Intent(getApplicationContext(), FriendLoadingActivity.class);
        intent.putExtra("user id", user id);
        intent.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
        startActivity(intent);
    } else if (id == group_list) {
        Intent intent=new Intent(getApplicationContext(), GroupLoadingActivity.class);
        intent.putExtra("user id", user id);
        intent.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
        startActivity(intent);
    } else if (id == R.id.logout) {
        //로그아웃
        UserManagement.requestLogout(new LogoutResponseCallback() {
            @Override
            public void onCompleteLogout() {
                Intent intent = new Intent(MainActivity.this, LoginActivity.class);
                startActivity(intent);
                overridePendingTransition(0,0);
                finish();
            }
        });
    }
}
});

```

```

    } else if (id == R.id.unlink) {
        onClickUnlink();

    }else if(id==R.id.main){

    }

    DrawerLayout drawer = (DrawerLayout) findViewById(R.id.drawer_layout);
    drawer.closeDrawer(GravityCompat.START);
    return true;
}

@Override
public void onMapClick(LatLng latLng) {
    //현재 지도에 마커가 없으면 클릭한 위치에 마커를 생성하고 마커가 있으면 기존 마커를 삭제하고 새로 클릭한
    위치에 마커를 생성
    if(currentmarker!=null){
        currentmarker.remove();
        currentmarker=map.addMarker(new
MarkerOptions().position(latLng).icon(BitmapDescriptorFactory.defaultMarker(BitmapDescriptorFactory.HUE_BLUE)));
    }else{
        currentmarker=map.addMarker(new
MarkerOptions().position(latLng).icon(BitmapDescriptorFactory.defaultMarker(BitmapDescriptorFactory.HUE_BLUE)));
    }
    lat=latLng.latitude;
    lng=latLng.longitude;
    currentpoint=lat.toString()+lng.toString();
}

@Override
public boolean onMarkerClick(Marker marker) {
    //마커가 클릭하면 그룹에서 가져온 마커리스트중 일치하는 마커의 정보를 받아와 정보창을 띄운다.
    LatLng position = marker.getPosition();
    Double lat = position.latitude;
    Double lng = position.longitude;
    String cp = Double.toString(lat) + Double.toString(lng);
    int i;
    for (i = 0; i < marker_list.size(); i++) {
        if (marker_list.get(i).getMarkerId().equals(cp)) {
            flag = true;
            this.name_place = marker_list.get(i).getPlace_name();
            this.writer_place = marker_list.get(i).getWriter();
            this.content_place = marker_list.get(i).getPlace_content();
            marker.showInfoWindow();
            break;
        } else {
            flag = false;
        }
    }
    return false;
}

//정보창을 길게 클릭하면 마커를 삭제하는 확인창이 팝업된다.
@Override
public void onInfoWindowLongClick(final Marker marker) {
    AlertDialog.Builder alert = new AlertDialog.Builder(this);
    alert.setTitle("마커 삭제");
    alert.setMessage("마커를 삭제 하시겠습니까?");
    alert.setPositiveButton("확인", new DialogInterface.OnClickListener() {
        @Override
        public void onClick(DialogInterface dialog, int which) {
            marker.remove();

```

```

        LatLng latLng=marker.getPosition();
        String lat= Double.toString(latLng.latitude);
        String lng=Double.toString(latLng.longitude);
        String markerid=lat+lng;
        deleteMarker(markerid);
    }

});
alert.setNegativeButton("취소", new DialogInterface.OnClickListener() {
    @Override
    public void onClick(DialogInterface arg0, int arg1) {

    }

});
alert.show();
}

@Override
public void onInfoWindowClick(Marker marker) {
    marker.hideInfoWindow();
}

@Override
//뒤로가기 버튼을 누르면 실행되는 동작
public void onBackPressed() {
    DrawerLayout drawer = (DrawerLayout) findViewById(R.id.drawer_layout);
    if (drawer.isDrawerOpen(GravityCompat.START)) {
        drawer.closeDrawer(GravityCompat.START);
    } else {
        super.onBackPressed();
    }
}

public boolean onCreateOptionsMenu(Menu menu) {
    getMenuInflater().inflate(R.menu.menu_item, menu);
    return true;
}

//우측상단의 검색버튼을 클릭하면 실행되는 동작
@Override
public boolean onOptionsItemSelected(MenuItem item) {
    int id=item.getItemId();
    if(id==R.id.action_search) {
        try {
            //검색버튼을 클릭하면 구글의 장소검색 자동완성 인텐트를 실행한다.
            Intent intent =
                new PlaceAutocomplete.IntentBuilder(PlaceAutocomplete.MODE_OVERLAY)
                    .build(this);
            startActivityForResult(intent, PLACE_AUTOCOMPLETE_REQUEST_CODE);
        } catch (GooglePlayServicesRepairableException e) {
            // TODO: Handle the error.
        } catch (GooglePlayServicesNotAvailableException e) {
            // TODO: Handle the error.
        }
    }
    return false;
}

@Override

//startForActivityResult를 통해 실행한 액티비티에서 받아오는 정보
protected void onActivityResult(int requestCode, int resultCode, Intent data) {

```

`super.onActivityResult(requestCode, resultCode, data);`
//장소검색 자동완성 인텐트에서 클릭한 장소에대한 위치값을 받아와 마커를 생성하고 그 위치로 줌되도록 한다.

```
if (requestCode == PLACE_AUTOCOMPLETE_REQUEST_CODE) {
    if (resultCode == RESULT_OK) {
        Place place = PlaceAutocomplete.getPlace(this, data);
        LatLng point=place.getLatLng();
        map.animateCamera(CameraUpdateFactory.newLatLngZoom(point, 17));
        if(currentmarker!=null){
            currentmarker.remove();
            currentmarker=map.addMarker(new MarkerOptions().position(point));
        }else{
            currentmarker=map.addMarker(new MarkerOptions().position(point));
        }
        manager.removeUpdates(gpsListener);
    } else if (resultCode == PlaceAutocomplete.RESULT_ERROR) {
    } else if (resultCode == RESULT_CANCELED) {
    }
}
```

//플로팅 버튼을 눌러 마커 추가하기 또는 마커가져오기에서 선택한 그룹의 마커들을 가져와 리스트에 저장한다.

```
if(requestCode==PLACE_GETMARKER_REQUEST_CODE){
    if(resultCode==RESULT_OK){
        for(int i=0;i<markers.size();i++){
            markers.get(i).remove();
        }
        marker_list=(ArrayList<Marker Info>)data.getSerializableExtra("markers");
        for(int i=0; i<marker_list.size();i++){
            Double lat=marker_list.get(i).getLat();
            Double lng=marker_list.get(i).getLng();
            Marker m = map.addMarker(new MarkerOptions().position(new
            LatLng(lat, lng)).icon(BitmapDescriptorFactory.defaultMarker(BitmapDescriptorFactory.HUE_GREEN)));
            markers.add(m);
        }
    }
}

}
```

//로그인된 사용자 정보를 받아온다.

```
private void requestMe() {
    UserManagement.requestMe(new MeResponseCallback() {
        @Override
        public void onFailure(ErrorResult errorResult) {
            String message = "failed to get user info. msg=" + errorResult;
            Logger.d(message);

            //redirectLoginActivity();
        }

        @Override
        public void onSessionClosed(ErrorResult errorResult) {
            //redirectLoginActivity();
        }

        @Override
        public void onSuccess(UserProfile userProfile) {
            Logger.d("UserProfile : " + userProfile);
            nickname = userProfile.getNickname(); //닉네임
            userID = userProfile.getId(); //사용자 고유번호
            plmage = userProfile.getProfileImagePath(); //사용자 프로필 경로
            user_email=userProfile.getEmail();
            emailText.setText(user_email);
        }
    });
}
```

```

        nameText.setText(nickName);
        userID=String.valueOf(userID);
        LinkImage();
    }

    @Override
    public void onNotSignedUp() {
        //showSignup();
    }
});
}

//카카오톡에서 받아온 프로필이미지의 url 주소를 이용해 비트맵이미지를 얻어온다.
Handler handler = new Handler();
public void LinkImage() {
    if (pimage.equals("img")) ;
    else {
        //new ImageDownload().execute(pimage);
        // 인터넷 상의 이미지 보여주기

        // 1. 권한을 획득한다 (인터넷에 접근할수 있는 권한을 획득한다) - 매니페스트 파일
        // 2. Thread 에서 웹의 이미지를 가져오기
        // 3. 외부쓰레드에서 메인 UI 에 접근위해 Handler 사용

        //Thread t = new Thread(Runnable 객체 생성);
        Thread t = new Thread(new Runnable() {
            @Override
            public void run() { // 오래 걸릴 작업 구현
                // TODO Auto-generated method stub
                try {
                    final URL url = new URL(pimage);
                    InputStream is = url.openStream();
                    final Bitmap bm = BitmapFactory.decodeStream(is);
                    handler.post(new Runnable() {

                        @Override
                        public void run() { // 화면에 그려줄 작업
                            profileImage.setImageBitmap(bm);
                            Log.d("imageurl",String.valueOf(url));
                        }
                    });
                    profileImage.setImageBitmap(bm); //비트맵 객체로 보여주기
                } catch (Exception e) {
                }
            }
        });
        t.start();
    }
}

//탈퇴하기
private void onClickUnlink() {
    final String appendMessage = getString(R.string.com_kakao_confirm_unlink);
    new AlertDialog.Builder(this)
        .setMessage(appendMessage)
        .setPositiveButton(getString(R.string.com_kakao_ok_button),
            new DialogInterface.OnClickListener() {
                @Override
                public void onClick(DialogInterface dialog, int which) {
                    UserManagement.requestUnlink(new UnLinkResponseCallback() {
                        @Override
                        public void onFailure(ErrorResult errorResult) {
                            Logger.e(errorResult.toString());
                        }
                    });
                }
            })
        .show();
}

```

```

    }

    @Override
    public void onSessionClosed(ErrorResult errorResult) {
        //redirectLoginActivity();
    }

    @Override
    public void onNotSignedUp() {
        //redirectSignupActivity();
    }

    @Override
    public void onSuccess(Long userId) {
        deleteuser(userId);
        Toast.makeText(getApplicationContext(), "탈퇴되었습니다.",
Toast.LENGTH_LONG).show();

        //redirectLoginActivity();
        Intent intent = new Intent(getApplicationContext(),
LoginActivity.class);

        startActivity(intent);
        overridePendingTransition(0,0);
        finish();
    }
    });
    dialog.dismiss();

    }
    })
    .setNegativeButton(getString(R.string.com_kakao_cancel_button),
        new DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialog, int which) {
                dialog.dismiss();
            }
        })
        .show();

}

//탈퇴하기를 클릭하고 onClickUnlink 메소드가 실행되면 데이터베이스에서 사용자 정보를 삭제한다.
public void deleteuser(long userId){
    try{
        PHPRequest request=new PHPRequest("http://180.71.13.212:8181/delete.php");
        String result=request.PhDelete(String.valueOf(userId));
        if(result.equals("1")){
            Toast.makeText(getApplicationContext(), "탈퇴되었습니다.",Toast.LENGTH_LONG).show();
        }else if(result.equals("-1")){
        }
    }catch (MalformedURLException e){
        e.printStackTrace();
    }
}

//GPSListener 구현
public class GPSListener implements LocationListener {
    public void onLocationChanged(Location location){
        melatitude=location.getLatitude();
        melongitude=location.getLongitude();
        mepoint=melatitude.toString()+melongitude.toString();
        showCurrentLocation(melatitude, melongitude);
    }
    @Override
    public void onProviderEnabled(String provider) {}

```

```

@Override
public void onProviderDisabled(String provider) {}
@Override
public void onStatusChanged(String provider, int status, Bundle extras) {}
}

public void showCurrentLocation(Double latitude, Double longitude){
    LatLng curPoint=new LatLng(latitude, longitude);
    map.animateCamera(CameraUpdateFactory.newLatLngZoom(curPoint, 17));
    map.setMapType(GoogleMap.MAP_TYPE_NORMAL);
}

private void startLocationService(){
    manager=(LocationManager) getSystemService(Context.LOCATION_SERVICE);
    gpsListener=new GPSTListener();
    long minTime=10000;
    float minDistance=0;
    try {
        manager.requestLocationUpdates(
            LocationManager.GPS_PROVIDER, //위치 정보 확인 방법 설정
            minTime, // 위치 정보 갱신 시간 설정
            minDistance, //위치 정보 갱신을 위한 최소 이동거리 설정
            gpsListener); //위치가 변동될 때마다 위치 정보 갱신을 위한 리스너 설정
        manager.requestLocationUpdates(
            LocationManager.NETWORK_PROVIDER,
            minTime,
            minDistance,
            gpsListener);
        Location lastLocation = manager.getLastKnownLocation(LocationManager.GPS_PROVIDER);
        if (lastLocation != null) {
            Double latitude = lastLocation.getLatitude();
            Double longitude = lastLocation.getLongitude();
        }
    } catch (SecurityException ex) {
        ex.printStackTrace();
    }
}

//카메라가 움직이면 gpsListener의 Update를 제거한다.
@Override
public void onCameraMoveStarted(int reason) {
    if(reason== GoogleMap.OnCameraMoveStartedListener.REASON_GESTURE){
        manager.removeUpdates(gpsListener);
    }
}

@Override
public void onConnectionFailed(@NonNull ConnectionResult connectionResult) {
}

//마커를 클릭했을때 나타나는 정보창을 구현한다.
@Override
public View getInfoWindow(Marker marker) {
    mView=getLayoutInflater().inflate(R.layout.information,null);
    place_name=(TextView)mView.findViewById(R.id.place_name);
    place_writer=(TextView)mView.findViewById(R.id.place_writer);
    place_content=(TextView)mView.findViewById(R.id.place_content);
    place_name.setText(name_place);
    place_writer.setText(writer_place);
    place_content.setText(content_place);
    if(flag==true){

```



```

        return mView;
    }else {
        return null;
    }
}

@Override
public View getInfoContents(Marker marker) {
    return null;
}

//마커를 데이터베이스에서 삭제한다.
public void deletemarker(String markerid){
    try{
        PHPRequest request=new PHPRequest("http://180.71.13.212:8181/deletemarker.php");
        String result=request.PhPdeletemarker(String.valueOf(markerid));
        if(result.equals("1")){
            Toast.makeText(getApplicationContext(),"마커를 제거했습니다!",Toast.LENGTH_LONG).show();
        }else if(result.equals("-1")){
            Toast.makeText(getApplicationContext(),"Fail!",Toast.LENGTH_LONG).show();
        }
    }catch (MalformedURLException e){
        e.printStackTrace();
    }
}
}
}

```

4) 마커 추가 또는 가져오기 화면

■ 레이아웃 구현

화면(UI)	설명
	<ul style="list-style-type: none"> - 메인에서 플로팅버튼을 클릭하면 화면과 같이 액티비티가 팝업되고 마커추가하기, 마커가져오기의 탭으로 이루어져 있다. - 마커추가하기에는 장소명과 내용을 입력하는 칸이 있고 다음으로 넘어가면 마커를 추가할 그룹을 선택하는 화면이 나온다. -마커가져오기에는 그룹을 선택하여 그룹에 속한 마커들을 지도상으로 불러올수있다.
activity_mainpopup.xml	
<pre> <?xml version="1.0" encoding="utf-8"?> <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android" xmlns:app="http://schemas.android.com/apk/res-auto" xmlns:tools="http://schemas.android.com/tools" android:layout_width="310dp" android:layout_height="380dp" tools:context="com.example.kimhyju.place.MainpopupActivity" android:weightSum="1"> </pre>	

```

<TabHost
    android:id="@+id/tab_host_main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_weight="0.77"
    tools:layout_editor_absoluteX="105dp"
    tools:layout_editor_absoluteY="122dp">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical">

        <TabWidget
            android:id="@android:id/tabs"
            android:layout_width="match_parent"
            android:layout_height="wrap_content" />

        <FrameLayout
            android:id="@android:id/tabcontent"
            android:layout_width="match_parent"
            android:layout_height="match_parent">

            <LinearLayout
                android:id="@+id/tab1_main"
                android:layout_width="match_parent"
                android:layout_height="match_parent"
                android:orientation="vertical">

                <FrameLayout
                    android:layout_width="match_parent"
                    android:layout_height="match_parent"
                    android:layout_weight="2"
                    android:visibility="visible">

                    <LinearLayout
                        android:id="@+id/writecontent"
                        android:layout_width="match_parent"
                        android:layout_height="match_parent"
                        android:orientation="vertical"
                        android:visibility="visible">

                        <EditText
                            android:id="@+id/place_name"
                            android:layout_width="match_parent"
                            android:layout_height="match_parent"
                            android:layout_marginBottom="8dp"
                            android:layout_marginLeft="16dp"
                            android:layout_marginRight="16dp"
                            android:layout_marginTop="16dp"
                            android:layout_weight="2"
                            android:background="@drawable/textbox"
                            android:cursorVisible="true"
                            android:ems="10"
                            android:hint="장소명"
                            android:inputType="textPersonName"
                            android:textColorHint="@color/place_autocomplete_search_hint"
                            app:layout_constraintLeft_toLeftOf="parent"
                            app:layout_constraintRight_toRightOf="parent"
                            app:layout_constraintTop_toTopOf="parent" />

                        <EditText

```

```

        android:id="@+id/place_content"
        style="@android:style/Widget.DeviceDefault.EditText"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_marginBottom="8dp"
        android:layout_marginLeft="16dp"
        android:layout_marginRight="16dp"
        android:layout_marginTop="8dp"
        android:layout_weight="1"
        android:background="@drawable/textbox"
        android:cursorVisible="true"
        android:ems="10"
        android:gravity="top"
        android:hint="내용"
        android:inputType="textMultiLine"
        android:selectAllOnFocus="false"
        android:singleLine="false"
        android:textColor="#000000"
        android:textColorHint="@color/place_autocomplete_search_hint"
        app:layout_constraintBottom_toTopOf="@+id/back"
        app:layout_constraintLeft_toLeftOf="@+id/place_name"
        app:layout_constraintTop_toBottomOf="@+id/place_name"
        app:layout_constraintVertical_bias="0.0" />

```

<LinearLayout

```

        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_weight="2"
        android:orientation="horizontal">

```

<Button

```

        android:id="@+id/mainpop_cancel"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_marginBottom="16dp"
        android:layout_marginLeft="16dp"
        android:layout_marginRight="8dp"
        android:layout_marginTop="8dp"
        android:layout_weight="2"
        android:background="@drawable/roundbutton_white"
        android:text="닫기"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintHorizontal_bias="1.0"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent" />

```

<Button

```

        android:id="@+id/mainpop_next"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_marginBottom="16dp"
        android:layout_marginLeft="8dp"
        android:layout_marginRight="16dp"
        android:layout_marginTop="8dp"
        android:layout_weight="2"
        android:background="@drawable/roundbutton_white"
        android:text="다음"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintHorizontal_bias="1.0"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent" />

```

</LinearLayout>

```

</LinearLayout>

<LinearLayout
    android:id="@+id/groupchoice"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:visibility="invisible">

    <ListView
        android:id="@+id/check_getgroup_list"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_margin="1dp"
        android:layout_weight="1"
        android:choiceMode="singleChoice" />

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_weight="4"
        android:orientation="horizontal">

        <Button
            android:id="@+id/mainpop_back"
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:layout_marginBottom="16dp"
            android:layout_marginLeft="16dp"
            android:layout_marginRight="8dp"
            android:layout_marginTop="8dp"
            android:layout_weight="2"
            android:background="@drawable/roundbutton_white"
            android:text="뒤로"
            app:layout_constraintBottom_toBottomOf="parent"
            app:layout_constraintHorizontal_bias="1.0"
            app:layout_constraintLeft_toLeftOf="parent"
            app:layout_constraintRight_toRightOf="parent" />

        <Button
            android:id="@+id/mainpop_confirm"
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:layout_marginBottom="16dp"
            android:layout_marginLeft="8dp"
            android:layout_marginRight="16dp"
            android:layout_marginTop="8dp"
            android:layout_weight="2"
            android:background="@drawable/roundbutton_white"
            android:text="확인"
            app:layout_constraintBottom_toBottomOf="parent"
            app:layout_constraintHorizontal_bias="1.0"
            app:layout_constraintLeft_toLeftOf="parent"
            app:layout_constraintRight_toRightOf="parent" />
    </LinearLayout>

</LinearLayout>

</FrameLayout>

</LinearLayout>

<LinearLayout

```

```

        android:id="@+id/tab2_main"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical">

        <LinearLayout
            android:id="@+id/groupchoice1"
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:orientation="vertical">

            <ListView
                android:id="@+id/check_setgroup_list"
                android:layout_width="match_parent"
                android:layout_height="match_parent"
                android:layout_margin="1dp"
                android:layout_weight="1"
                android:choiceMode="singleChoice" />

            <LinearLayout
                android:layout_width="match_parent"
                android:layout_height="match_parent"
                android:layout_weight="4"
                android:orientation="horizontal">

                <Button
                    android:id="@+id/mainpop_cancel1"
                    android:layout_width="match_parent"
                    android:layout_height="match_parent"
                    android:layout_marginBottom="16dp"
                    android:layout_marginLeft="16dp"
                    android:layout_marginRight="8dp"
                    android:layout_marginTop="8dp"
                    android:layout_weight="2"
                    android:background="@drawable/roundbutton_white"
                    android:text="닫기"
                    app:layout_constraintBottom_toBottomOf="parent"
                    app:layout_constraintHorizontal_bias="1.0"
                    app:layout_constraintLeft_toLeftOf="parent"
                    app:layout_constraintRight_toRightOf="parent" />

                <Button
                    android:id="@+id/mainpop_confirm1"
                    android:layout_width="match_parent"
                    android:layout_height="match_parent"
                    android:layout_marginBottom="16dp"
                    android:layout_marginLeft="8dp"
                    android:layout_marginRight="16dp"
                    android:layout_marginTop="8dp"
                    android:layout_weight="2"
                    android:background="@drawable/roundbutton_white"
                    android:text="확인"
                    app:layout_constraintBottom_toBottomOf="parent"
                    app:layout_constraintHorizontal_bias="1.0"
                    app:layout_constraintLeft_toLeftOf="parent"
                    app:layout_constraintRight_toRightOf="parent" />

            </LinearLayout>

        </LinearLayout>

    </LinearLayout>

</FrameLayout>

```

```
        </LinearLayout>  
    </TabHost>  
</LinearLayout>
```

■ 기능 구현

MainpopupActivity.java

```
package com.example.kimhyju.place;

import android.app.Activity;
import android.app.ProgressDialog;
import android.content.Intent;
import android.os.AsyncTask;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.view.Window;
import android.view.animation.Animation;
import android.view.animation.AnimationUtils;
import android.widget.AdapterView;
import android.widget.Button;
import android.widget.EditText;
import android.widget.LinearLayout;
import android.widget.ListView;
import android.widget.TabHost;
import android.widget.Toast;

import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

import java.io.BufferedReader;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.net.HttpURLConnection;
import java.net.MalformedURLException;
import java.net.URL;
import java.util.ArrayList;

public class MainpopupActivity extends Activity implements View.OnClickListener{
    TabHost tabHost;
    private static String TAG="phptest_listActivity";
    private static final String TAG_JSON="kimhyju";
    private static final String TAG_GROUPNAME="group_name";
    String mJsonString;
    ListView getgroup_list,setgroup_list;
    String userID;
    String address, address1;
    Double lat,lng;
    String currentpoint;
    LinearLayout writecontent, groupchoice;
    Animation transRightMid, transMidRight, transLeftMid, transMidLeft;
    Button mainpop_cancel,mainpop_next,mainpop_back,mainpop_confirm,mainpop_cancel1,mainpop_confirm1;
    EditText placename, placecontent;
    String place_name,place_content;
    String username;
    ArrayList<String> groupname_list;
    ArrayAdapter<String> adapter;
    ArrayList<MarkerInfo> marker_list;
    String gn;
    GetMarker task1;
    boolean page =false;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
```



```

requestWindowFeature(Window.FEATURE_NO_TITLE);
setContentView(R.layout.activity_mainpopup);
NetworkUtil.setNetworkPolicy();
tabHost=(TabHost)findViewById(R.id.tab_host_main);
getgroup_list=(ListView)findViewById(R.id.check_getgroup_list);
setgroup_list=(ListView)findViewById(R.id.check_setgroup_list);
writecontent=(LinearLayout)findViewById(R.id.writecontent);
groupchoice=(LinearLayout)findViewById(R.id.groupchoice);
transLeftMid= AnimationUtils.loadAnimation(this,R.anim.translate_leftmid);
transMidLeft= AnimationUtils.loadAnimation(this,R.anim.translate_midleft);
transRightMid= AnimationUtils.loadAnimation(this,R.anim.translate_rightmid);
transMidRight= AnimationUtils.loadAnimation(this,R.anim.translate_midright);
mainpop_cancel=(Button)findViewById(R.id.mainpop_cancel);
mainpop_next=(Button)findViewById(R.id.mainpop_next);
mainpop_back=(Button)findViewById(R.id.mainpop_back);
mainpop_confirm=(Button)findViewById(R.id.mainpop_confirm);
mainpop_cancel1=(Button)findViewById(R.id.mainpop_cancel1);
mainpop_confirm1=(Button)findViewById(R.id.mainpop_confirm1);
placename=(EditText)findViewById(R.id.place_name);
placecontent=(EditText)findViewById(R.id.place_content);
mainpop_cancel.setOnClickListener(this);
mainpop_next.setOnClickListener(this);
mainpop_back.setOnClickListener(this);
mainpop_confirm.setOnClickListener(this);
mainpop_cancel1.setOnClickListener(this);
mainpop_confirm1.setOnClickListener(this);
groupname_list=new ArrayList<>();

SlidingPageAnimationListener animationListener=new SlidingPageAnimationListener();
tabHost.setup();
TabHost.TabSpec tab1 = tabHost.newTabSpec("1").setContent(R.id.tab1_main).setIndicator("마커
추가하기");
TabHost.TabSpec tab2 = tabHost.newTabSpec("2").setContent(R.id.tab2_main).setIndicator("마커
가져오기");
tabHost.addTab(tab1);
tabHost.addTab(tab2);
adapter=new
ArrayAdapter<String>(getApplicationContext(),android.R.layout.simple_list_item_single_choice,groupname
_list);

userID=getIntent().getStringExtra("user id");
lat=getIntent().getDoubleExtra("lat",0);
lng=getIntent().getDoubleExtra("lng",0);
username=getIntent().getStringExtra("username");
currentpoint=getIntent().getStringExtra("currentpoint");
address="http://180.71.13.212:8181/getgroup.php?user id="+userID;
GetData task=new GetData();
task.execute(address);
marker_list=new ArrayList<>();

transLeftMid.setAnimationListener(animationListener);
transMidRight.setAnimationListener(animationListener);
transRightMid.setAnimationListener(animationListener);
transMidLeft.setAnimationListener(animationListener);
}

@Override
public void onClick(View v) {
    switch(v.getId()){
        case R.id.mainpop_cancel:
        case R.id.mainpop_cancel1:
            finish();
            break;
    }
}

```

```

        case R.id.mainpop_next: //마커 추가하기 탭에서 다음 버튼을 누르면 슬라이드되면서 그룹선택
        페이지가 나온다.
            place_name=placename.getText().toString();
            place_content=placecontent.getText().toString();
            if(place_name.isEmpty()==true||place_content.isEmpty()==true){
                Toast.makeText(getApplicationContext(),"제목과 내용을
                입력해주세요!",Toast.LENGTH_LONG).show();
            }else {
                groupchoice.setVisibility(View.VISIBLE);
                groupchoice.startAnimation(transRightMid);
                writecontent.startAnimation(transMidLeft);
            }
            break;
        case R.id.mainpop_back:
            writecontent.setVisibility(View.VISIBLE);
            groupchoice.startAnimation(transMidRight);
            writecontent.startAnimation(transLeftMid);
            break;
        case R.id.mainpop_confirm: //마커 추가하기
            place_name=placename.getText().toString();
            place_content=placecontent.getText().toString();
            int getposition=getgroup_list.getCheckedItemPosition();
            gn=groupname_list.get(getposition);
            addmarker();
            address1="http://180.71.13.212:8181/getmarker.php?name_group="+gn;
            task1=new GetMarker();
            task1.execute(address1);
            break;
        case R.id.mainpop_confirm1: //마커 가져오기
            int setposition=setgroup_list.getCheckedItemPosition();
            gn=groupname_list.get(setposition);
            address1="http://180.71.13.212:8181/getmarker.php?name_group="+gn;
            task1=new GetMarker();
            task1.execute(address1);
            break;
    }
}

//마커를 데이터베이스에 등록한다.
public void addmarker(){
    try{
        PHPRequest request=new PHPRequest("http://180.71.13.212:8181/addmarker.php");
        String
        result=request.Phpaddmarker(String.valueOf(currentpoint),String.valueOf(lat),String.valueOf(lng),String
        g.valueOf(place_name),String.valueOf(gn),String.valueOf(place_content),String.valueOf(user name),String
        .valueOf(userID));
        if(result.equals("1")){
            Toast.makeText(getApplicationContext(),"마커를
            등록했습니다!",Toast.LENGTH_LONG).show();
        }else if(result.equals("-1")){
            Toast.makeText(getApplicationContext(),"마커 등록에
            실패했습니다!",Toast.LENGTH_LONG).show();
        }
    }catch (MalformedURLException e){
        e.printStackTrace();
    }
}

//그룹목록을 데이터베이스에서 가져오는 클래스
private class GetData extends AsyncTask<String, Void, String> {
    String errorString = null;

```

```

@Override
protected void onPreExecute() {
    super.onPreExecute();
}

@Override
protected void onPostExecute(String result) {
    super.onPostExecute(result);
    mJsonString=result;
    try {
        showResult();
    }catch(Exception e){
        Log.d("error", "에러발생");
    }
}

@Override
protected String doInBackground(String... params) {

    String serverURL = params[0];
    try {

        URL url = new URL(serverURL);
        HttpURLConnection httpURLConnection = (HttpURLConnection) url.openConnection();
        httpURLConnection.setReadTimeout(5000);
        httpURLConnection.setConnectTimeout(5000);
        httpURLConnection.connect();
        int responseStatusCode = httpURLConnection.getResponseCode();
        Log.d(TAG, "response code - " + responseStatusCode);
        InputStream inputStream;
        if(responseStatusCode == HttpURLConnection.HTTP_OK) {
            inputStream = httpURLConnection.getInputStream();
        }
        else{
            inputStream = httpURLConnection.getErrorStream();
        }
        InputStreamReader inputStreamReader = new InputStreamReader(inputStream, "UTF-8");
        BufferedReader bufferedReader = new BufferedReader(inputStreamReader);
        StringBuilder sb = new StringBuilder();
        String line;
        while((line = bufferedReader.readLine()) != null){
            sb.append(line);
        }
        bufferedReader.close();
        return sb.toString().trim();
    } catch (Exception e) {
        Log.d(TAG, "InsertData: Error ", e);
        errorString = e.toString();
        return null;
    }
}

private void showResult(){
    try {
        JSONObject jsonObject = new JSONObject(mJsonString);
        JSONArray jsonArray = jsonObject.getJSONArray( TAG_JSON );

        for(int i=0;i<javascriptArray.length();i++){
            JSONObject item = jsonArray.getJSONObject(i);
            String groupname = item.getString( TAG_GROUPNAME );
            groupname_list.add(groupname);
        }
        getgroup_list.setAdapter(adapter);
        setgroup_list.setAdapter(adapter);
    }
}

```

```

    } catch (JSONException e) {
        Log.d(TAG, "showResult : ", e);
    }
}

```

//마커추가하기 탭에서 화면을 넘기는 애니메이션 클래스

```

private class SlidingPageAnimationListener implements Animation.AnimationListener{
    @Override
    public void onAnimationStart(Animation animation) {
    }
    @Override
    public void onAnimationEnd(Animation animation) {
        if(page){
            groupchoice.setVisibility(View.INVISIBLE);
            page=false;
        }else{
            writecontent.setVisibility(View.INVISIBLE);
            page=true;
        }
    }
    @Override
    public void onAnimationRepeat(Animation animation) {
    }
}

```

//그룹에 속한 마커를 가져오는 클래스

```

private class GetMarker extends AsyncTask<String, Void, String> {
    String errorString = null;
    ProgressDialog progressDialog=new ProgressDialog(MainpopupActivity.this);
    @Override
    protected void onPreExecute() {
        progressDialog.setProgressStyle(ProgressDialog.STYLE_SPINNER);
        progressDialog.setMessage("로딩중입니다...");
        progressDialog.show();
        super.onPreExecute();
    }
}

```

```

    @Override
    protected void onPostExecute(String result) {
        super.onPostExecute(result);
        mJsonString=result;
        try {
            showMarker();
        }catch(Exception e){
            Log.d("error", "에러발생");
        }
    }
}

```

//메인 팝업 액티비티에서 마커를 가져와 ArrayList 에 저장하고 그 리스트를 intent 로 메인액티비티에 넘겨준다.

```

        progressDialog.dismiss();
        Intent intent=new Intent(MainpopupActivity.this,MainActivity.class);
        intent.putExtra("markers",marker_list);
        setResult(RESULT_OK,intent);
        finish();
    }
}

```

```

    @Override
    protected String doInBackground(String... params) {
        String serverURL = params[0];
        try {
            URL url = new URL(serverURL);
            HttpURLConnection httpURLConnection = (HttpURLConnection) url.openConnection();
            httpURLConnection.setReadTimeout(5000);

```

```

        httpURLConnection.setConnectTimeout(5000);
        httpURLConnection.connect();
        int responseStatusCode = httpURLConnection.getResponseCode();
        Log.d(TAG, "response code - " + responseStatusCode);
        InputStream inputStream;
        if(responseStatusCode == HttpURLConnection.HTTP_OK) {
            inputStream = httpURLConnection.getInputStream();
        }
        else{
            inputStream = httpURLConnection.getErrorStream();
        }
        InputStreamReader inputStreamReader = new InputStreamReader(inputStream, "UTF-8");
        BufferedReader bufferedReader = new BufferedReader(inputStreamReader);
        StringBuilder sb = new StringBuilder();
        String line;
        while((line = bufferedReader.readLine()) != null){
            sb.append(line);
        }
        bufferedReader.close();
        return sb.toString().trim();
    } catch (Exception e) {
        Log.d(TAG, "InsertData: Error ", e);
        errorString = e.toString();
        return null;
    }
}

}

private void showMarker(){
    try {
        JSONObject jsonObject = new JSONObject(mJsonString);
        JSONArray jsonArray = jsonObject.getJSONArray(TAG_JSON);
        for(int i=0; i<javascriptArray.length(); i++){
            JSONObject item = jsonArray.getJSONObject(i);
            String markerId=item.getString("marker id");
            Double lat=item.getDouble("lat");
            Double lng=item.getDouble("lng");
            String name_place=item.getString("name_place");
            String content=item.getString("content");
            String name_user=item.getString("name_user");
            marker_list.add(new MarkerInfo(markerId,lat,lng,name_place,name_user,content));
        }
    } catch (JSONException e) {
        Log.d(TAG, "showResult : ", e);
    }
}
}
}

```

5) 그룹목록 로딩 액티비티

■ 레이아웃 구현

화면(UI)	설명
화면없음	<ul style="list-style-type: none"> - 메인화면은 타이틀과 리스트뷰 그리고 New Memo, Close 버튼으로 구성 - 저장된 메모를 보여주는 리스트뷰는 어찌구...저찌구 <p style="color: red;">레이아웃 설명하세요.</p>
레이아웃없음	
레이아웃없음	

■ 기능 구현

GroupLoadingActivity.java

```
package com.example.kimhyju.place;

import android.app.Activity;
import android.content.Intent;
import android.os.AsyncTask;
import android.os.Bundle;
import android.support.annotation.Nullable;
import android.util.Log;

import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

import java.io.BufferedReader;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.net.HttpURLConnection;
import java.net.URL;
import java.util.ArrayList;

//바로 그룹목록으로 넘어가면 그룹목록을 데이터베이스에 받아오기전에 화면이 출력되어 간혹 그룹목록이
//화면에 나타나지 않는 현상이
//발생함으로써 GroupLoadingActivity를 구현해 그룹목록을 데이터베이스에서 받아오기가 완료되면 intent로
//그룹목록을 넘겨주고
//GroupActivity가 실행되도록 하였다.
public class GroupLoadingActivity extends Activity {
    private static String TAG="phptest_listActivity";
    private static final String TAG_JSON="kimhyju";
    ArrayList<GroupItem> list;
    String mJsonString;
    String userId;
    String address;
    @Override
    protected void onCreate(@Nullable Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        NetworkUtil.setNetworkPolicy();
        list=new ArrayList<>();
        userId=getIntent().getStringExtra("user id");
        address="http://180.71.13.212:8181/getgroup.php?user id="+userId;
        GetData task=new GetData();
        task.execute(address);
    }

    //그룹목록을 데이터베이스에서 받아와서 그룹액티비티로 보내기위해 리스트에 저장한다
```

```

private class GetData extends AsyncTask<String, Void, String> {
    String errorString = null;
    @Override
    protected void onPreExecute() {
        super.onPreExecute();
    }

    //그룹을 받아와서 리스트에 모두 저장하면 그룹액티비티에 리스트를 넘겨주고 그룹액티비티를
    실행한다
    @Override
    protected void onPostExecute(String result) {
        mJsonString=result;
        try {
            showResult();
        } catch (Exception e){

        }
        Intent intent=new Intent(GroupLoadingActivity.this,GroupActivity.class);
        intent.putExtra("list",list);
        intent.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
        startActivity(intent);
        finish();
        super.onPostExecute(result);
    }
    @Override
    protected String doInBackground(String... params) {
        String serverURL = params[0];
        try {
            URL url = new URL(serverURL);
            HttpURLConnection httpURLConnection = (HttpURLConnection) url.openConnection();
            httpURLConnection.setReadTimeout(5000);
            httpURLConnection.setConnectTimeout(5000);
            httpURLConnection.connect();
            int responseStatusCode = httpURLConnection.getResponseCode();
            Log.d(TAG, "response code - " + responseStatusCode);
            InputStream inputStream;
            if(responseStatusCode == HttpURLConnection.HTTP_OK) {
                inputStream = httpURLConnection.getInputStream();
            }
            else{
                inputStream = httpURLConnection.getErrorStream();
            }
            InputStreamReader inputStreamReader = new InputStreamReader(inputStream, "UTF-8");
            BufferedReader bufferedReader = new BufferedReader(inputStreamReader);
            StringBuilder sb = new StringBuilder();
            String line;
            while((line = bufferedReader.readLine()) != null){
                sb.append(line);
            }
            bufferedReader.close();
            return sb.toString().trim();
        } catch (Exception e) {
            Log.d(TAG, "InsertData: Error ", e);
            errorString = e.toString();
            return null;
        }
    }
}

private void showResult(){
    try {
        JSONObject jsonObject = new JSONObject(mJsonString);
        JSONArray jsonArray = jsonObject.getJSONArray(TAG_JSON);
        for(int i=0; i<jarray.length(); i++){

```


```

        JSONObject item = jsonArray.getJSONObject(i);
        String relationid=item.getString("relationid");
        String groupname = item.getString("group_name");
        GroupItem gi=new GroupItem(relationid, groupname);
        list.add(gi);
    }
} catch (JSONException e) {
    Log.d(TAG, "showResult : ", e);
}
}
}

```

6) 그룹목록화면

■ 레이아웃 구현

화면(UI)	설명
	<ul style="list-style-type: none"> - 메뉴버튼, 플로팅버튼, 그룹리스트로 이루어져있다. - 리스트는 직접레이아웃을 제작하여 어댑터로 구현하였다.
activity_group.xml	
<pre> <?xml version="1.0" encoding="utf-8" ?> <android.support.v4.widget.DrawerLayout xmlns:android="http://schemas.android.com/apk/res/android" xmlns:app="http://schemas.android.com/apk/res-auto" xmlns:tools="http://schemas.android.com/tools" android:id="@+id/drawer_layout" android:layout_width="match_parent" android:layout_height="match_parent" android:fitsSystemWindows="true" tools:openDrawer="start"> <include layout="@layout/app_bar_group" android:layout_width="match_parent" android:layout_height="match_parent" /> <android.support.design.widget.NavigationView android:id="@+id/nav_view" android:layout_width="wrap_content" android:layout_height="match_parent" android:layout_gravity="start" android:fitsSystemWindows="true" app:headerLayout="@layout/nav_header_main2" app:menu="@menu/activity_main2_drawer" /> </android.support.v4.widget.DrawerLayout> </pre>	

app_bar_group.xml

```

<?xml version="1.0" encoding="utf-8" ?>
<android.support.design.widget.CoordinatorLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="com.example.kimhyju.place.GroupActivity">

    <android.support.design.widget.AppBarLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:theme="@style/AppTheme.AppBarOverlay">

        <android.support.v7.widget.Toolbar
            android:id="@+id/toolbar_group"
            android:layout_width="match_parent"
            android:layout_height="?attr/actionBarSize"
            android:background="?attr/colorPrimary"
            app:popupTheme="@style/AppTheme.PopupOverlay" />

    </android.support.design.widget.AppBarLayout>

    <include layout="@layout/content_group" />

    <android.support.design.widget.FloatingActionButton
        android:id="@+id/group_fab"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="bottom|end"
        android:layout_margin="@dimen/fab_margin"
        app:backgroundTint="?attr/colorPrimary"
        app:srcCompat="@drawable/add" />

</android.support.design.widget.CoordinatorLayout>

```

content_group.xml

```

<?xml version="1.0" encoding="utf-8" ?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    app:layout_behavior="@string/appbar_scrolling_view_behavior"
    tools:context="com.example.kimhyju.place.GroupActivity"
    tools:showIn="@layout/app_bar_group">

    <ListView
        android:id="@+id/group_list"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_marginBottom="0dp"
        android:layout_marginLeft="0dp"
        android:layout_marginRight="0dp"
        android:layout_marginTop="0dp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

```

</LinearLayout>

■ 기능 구현

GroupActivity.java

```

package com.example.kimhyju.place;

import android.content.DialogInterface;
import android.content.Intent;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.os.Bundle;
import android.os.Handler;
import android.support.design.widget.FloatingActionButton;
import android.support.design.widget.NavigationView;
import android.support.v4.view.GravityCompat;
import android.support.v4.widget.DrawerLayout;
import android.support.v7.app.ActionBarDrawerToggle;
import android.support.v7.app.AlertDialog;
import android.support.v7.app.AppCompatActivity;
import android.support.v7.widget.PopupMenu;
import android.support.v7.widget.Toolbar;
import android.util.Log;
import android.view.MenuItem;
import android.view.View;
import android.widget.AdapterView;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.ListView;
import android.widget.TextView;
import android.widget.Toast;

import com.kakao.auth.Session;
import com.kakao.network.ErrorResult;
import com.kakao.usermgmt.UserManagement;
import com.kakao.usermgmt.callback.LogoutResponseCallback;
import com.kakao.usermgmt.callback.MeResponseCallback;
import com.kakao.usermgmt.callback.UnLinkResponseCallback;
import com.kakao.usermgmt.response.model.UserProfile;
import com.kakao.util.helper.log.Logger;

import java.io.InputStream;
import java.net.MalformedURLException;
import java.net.URL;
import java.util.ArrayList;

import static com.example.kimhyju.place.R.id. friend_list;

public class GroupActivity extends AppCompatActivity
    implements NavigationView.OnNavigationItemSelectedListener, View.OnClickListener{
    View nav_head_view;
    private ImageView profileImage;
    private TextView nameText;
    private Button confirmID;
    String nickName = ""; //닉네임
    long userID = 0; //사용자 고유번호
    String userid;
    String pImage = ""; //사용자 프로필 경로
    String user_email;
    TextView emailText;
    Toolbar toolbar;
    ArrayList<GroupItem> list;

```

```

ListView group_list;
GroupAdapter adapter;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_group);
    NetworkUtil.setNetworkPolicy();
    list=new ArrayList<>();
    list=(ArrayList<GroupItem>)getIntent().getSerializableExtra("list");
    NavigationView navigationView = (NavigationView) findViewById(R.id.nav_view);
    nav_head_view=navigationView.getHeaderView(0);
    profileImage=(ImageView)nav_head_view.findViewById(R.id.profileImage);
    nameText=(TextView)nav_head_view.findViewById(R.id.nameText1);
    emailText=(TextView)nav_head_view.findViewById(R.id.email);
    confirmID=(Button)nav_head_view.findViewById(R.id.confirmID);
    toolbar = (Toolbar) findViewById(R.id.toolbar_group);
    setSupportActionBar(toolbar);
    getSupportActionBar().setTitle("그룹목록");
    confirmID.setOnClickListener(this);
    group_list=(ListView)findViewById(R.id.group_list);
    adapter=new GroupAdapter(getApplicationContext(),R.layout.group_list,list);
    Session.getCurrentSession().checkAccessTokenInfo();//확인 차 한번 더 토큰확인

    //FloatingActionButton 을 누르면 그룹을 생성또는 가입하기 위해 GroupLoadingActivity 에서
    가져왔던 그룹리스트를 GrouppopupActivity 로 전달한다.
    FloatingActionButton fab = (FloatingActionButton) findViewById(R.id.group_fab);
    fab.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            Intent intent=new Intent(getApplicationContext(),GrouppopupActivity.class);
            intent.putExtra("user id",user id);
            intent.putExtra("list",list);
            startActivity(intent);
        }
    });

    DrawerLayout drawer = (DrawerLayout) findViewById(R.id.drawer_layout);
    ActionBarDrawerToggle toggle = new ActionBarDrawerToggle(
        this, drawer, toolbar, R.string.navigation_drawer_open,
        R.string.navigation_drawer_close);
    drawer.setDrawerListener(toggle);
    toggle.syncState();
    navigationView.setNavigationItemSelectedListener(this);
    requestMe();
    adapter.notifyDataSetChanged();
    group_list.setAdapter(adapter);

    //그룹목록을 길게 터치하면 삭제버튼이 나타난다. 삭제버튼을 누르면 그룹이 삭제된다.
    group_list.setOnItemLongClickListener(new AdapterView.OnItemLongClickListener() {
        @Override
        public boolean onItemLongClick(AdapterView<?> parent, View view, int position, long id) {
            PopupMenu popup=new PopupMenu(GroupActivity.this,view);
            getLayoutInflater().inflate(R.menu.popupmenu,popup.getMenu());
            final int index=position;
            popup.setOnMenuItemClickListener(new PopupMenu.OnMenuItemClickListener() {
                @Override
                public boolean onMenuItemClick(MenuItem item) {
                    if(item.getItemId()==R.id.action_delete){
                        String relationid=list.get(index).getRelationid();
                        Intent intent=new Intent(GroupActivity.this,GroupLoadingActivity.class);
                        deletegroup(relationid);
                        intent.putExtra("user id",user id);
                    }
                }
            });
            popup.show();
            return true;
        }
    });
}

```

```

        intent.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
        startActivity(intent);
        finish();
    }
    return false;
}
});
popup.show();
return false;
}
});
});

//뒤로가기버튼을 눌렀을때 메뉴가 열려있으면 메뉴가 닫히고 메뉴가 닫혀있으면 액티비티가 종료된다
@Override
public void onBackPressed() {
    DrawerLayout drawer = (DrawerLayout) findViewById(R.id.drawer_layout);
    if (drawer.isDrawerOpen(GravityCompat.START)) {
        drawer.closeDrawer(GravityCompat.START);
    } else {
        finish();
    }
}

@SuppressWarnings("StatementWithEmptyBody")
@Override
public boolean onNavigationItemSelected(MenuItem item) {
    // Handle navigation view item clicks here.
    int id = item.getItemId();
    item.setCheckable(false);
    if (id == R.id.friend_list) {
        Intent intent = new Intent(getApplicationContext(), FriendLoadingActivity.class);
        intent.putExtra("user id", user id);
        startActivity(intent);
        finish();
    } else if (id == R.id.group_list) {
    } else if (id == R.id.logout) {
        UserManagement.requestLogout(new LogoutResponseCallback() {
            @Override
            public void onCompleteLogout() {
                Intent intent = new Intent(GroupActivity.this, LoginActivity.class);
                startActivity(intent);
                overridePendingTransition(0, 0);
                finish();
            }
        });
    } else if (id == R.id.unlink) {
        onClickUnlink();
    } else if (id == R.id.main) {
        finish();
    }

    DrawerLayout drawer = (DrawerLayout) findViewById(R.id.drawer_layout);
    drawer.closeDrawer(GravityCompat.START);
    return true;
}

Handler handler = new Handler();
public void LinkImage() {

```

```

if (pimage.equals("img")) ;
else {
    //new ImageDownload().execute(pimage);
    // 인터넷 상의 이미지 보여주기

    // 1. 권한을 획득한다 (인터넷에 접근할수 있는 권한을 획득한다) - 매니페스트 파일
    // 2. Thread 에서 웹의 이미지를 가져오기
    // 3. 외부쓰레드에서 메인 UI 에 접근위해 Handler 사용

    //Thread t = new Thread(Runnable 객체 생성);
    Thread t = new Thread(new Runnable() {
        @Override
        public void run() { // 오래 걸릴 작업 구현
            // TODO Auto-generated method stub
            try {
                final URL url = new URL(pimage);
                InputStream is = url.openStream();
                final Bitmap bm = BitmapFactory.decodeStream(is);
                handler.post(new Runnable() {

                    @Override
                    public void run() { // 화면에 그려줄 작업
                        profileImage.setImageBitmap(bm);
                        Log.d("imageurl",String.valueOf(url));
                    }
                });
                profileImage.setImageBitmap(bm); //비트맵 객체로 보여주기
            } catch (Exception e) {
            }
        }
    });
    t.start();
}

private void onClickUnlink() {
    final String appendMessage = getString(R.string.com_kakao_confirm_unlink);
    new AlertDialog.Builder(this)
        .setMessage(appendMessage)
        .setPositiveButton(getString(R.string.com_kakao_ok_button),
            new DialogInterface.OnClickListener() {
                @Override
                public void onClick(DialogInterface dialog, int which) {
                    UserManagement.requestUnlink(new UnLinkResponseCallback() {
                        @Override
                        public void onFailure(ErrorResult errorResult) {
                            Logger.e(errorResult.toString());
                        }
                        @Override
                        public void onSessionClosed(ErrorResult errorResult) {
                            //redirectLoginActivity();
                        }
                        @Override
                        public void onNotSignedUp() {
                            //redirectSignupActivity();
                        }
                        @Override
                        public void onSuccess(Long userId) {
                            deleteUser(userId);
                            Toast.makeText(getApplicationContext(), "탈퇴되었습니다.",
                                Toast.LENGTH_LONG).show();

                            //redirectLoginActivity();
                            Intent intent = new Intent(getApplicationContext(),
                                LoginActivity.class);

```

```

        startActivity(intent);
        overridePendingTransition(0,0);
        finish();
    }
});
dialog.dismiss();
}
})
.setNegativeButton(getString(R.string.com_kakao_cancel_button),
    new DialogInterface.OnClickListener() {
        @Override
        public void onClick(DialogInterface dialog, int which) {
            dialog.dismiss();
        }
    }).show();
}

@Override
public void onClick(View v) {
    if(v.getId()==R.id.confirmID){
        Toast.makeText(getApplicationContext(),"당신의 ID 번호는
"+user ID+"입니다.",Toast.LENGTH_LONG).show();
    }
}

private void requestMe() {
    UserManagement.requestMe(new MeResponseCallback() {
        @Override
        public void onFailure(ErrorResult errorResult) {
            String message = "failed to get user info. msg=" + errorResult;
            Logger.d(message);
        }
        @Override
        public void onSessionClosed(ErrorResult errorResult) {

        }

        @Override
        public void onSuccess(UserProfile userProfile) {
            Logger.d("UserProfile : " + userProfile);
            nickname = userProfile.getNickname();//닉네임
            userID = userProfile.getId();//사용자 고유번호
            plmage = userProfile.getProfileImagePath();//사용자 프로필 경로
            user_email=userProfile.getEmail();
            emailText.setText(user_email);
            nameText.setText(nickname);
            user id=String.valueOf(userID);
            LinkImage();
        }
        @Override
        public void onNotSignedUp() {

        }
    });
}

//탈퇴하기
public void deleteuser(long user id){
    try{
        PHPRequest request=new PHPRequest("http://180.71.13.212:8181/delete.php");
        String result=request.PhDelete(String.valueOf(user id));
        if(result.equals("1")){
            Toast.makeText(getApplicationContext(),"Success!",Toast.LENGTH_LONG).show();
        }else if(result.equals("-1")){

```

```

        Toast.makeText(getApplicationContext(),"Fail!",Toast.LENGTH_LONG).show();
    }
} catch (MalformedURLException e){
    e.printStackTrace();
}
}

//
public void deletegroup(String relationid){
    try{
        PHPRequest request=new PHPRequest("http://180.71.13.212:8181/deletegroup.php");
        String result=request.PhpDeletegroup(String.valueOf(relationid));
        if(result.equals("1")){
            Toast.makeText(getApplicationContext(),"그룹에서
탈퇴했습니다!",Toast.LENGTH_LONG).show();
        }else if(result.equals("-1")){
            Toast.makeText(getApplicationContext(),"Fail!",Toast.LENGTH_LONG).show();
        }
    } catch (MalformedURLException e){
        e.printStackTrace();
    }
}
}
}

```

7) 그룹생성 또는 가입하기화면

■ 레이아웃 구현

화면(UI)	설명
	<ul style="list-style-type: none"> - 그룹목록화면에서 플로팅버튼을 클릭하면 그룹생성하기와 그룹 가입하기의 탭으로 이루어진 액티비티가 실행된다. - 그룹생성하기에는 그룹명과 비밀번호를 입력하는칸이 있다. - 그룹 가입하기도 똑 같은 레이아웃으로 구성되어있고 비밀번호가 틀리면 가입이 안된다.
activity_grouppopup.xml	
<pre> <?xml version="1.0" encoding="utf-8" ?> <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android" xmlns:app="http://schemas.android.com/apk/res-auto" xmlns:tools="http://schemas.android.com/tools" android:layout_width="310dp" android:layout_height="250dp" android:weightSum="1" tools:context="com.example.kimhyju.place.MainpopupActivity"> <TabHost </pre>	

```

android:id="@+id/tab_host_group"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:layout_weight="0.77"
tools:layout_editor_absoluteX="105dp"
tools:layout_editor_absoluteY="122dp">

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">

    <TabWidget
        android:id="@android:id/tabs"
        android:layout_width="match_parent"
        android:layout_height="wrap_content" />

    <FrameLayout
        android:id="@android:id/tabcontent"
        android:layout_width="match_parent"
        android:layout_height="match_parent">

        <LinearLayout
            android:id="@+id/tab1_group"
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:orientation="vertical">

            <EditText
                android:id="@+id/makegroup_name"
                android:layout_width="match_parent"
                android:layout_height="match_parent"
                android:layout_marginBottom="8dp"
                android:layout_marginLeft="16dp"
                android:layout_marginRight="16dp"
                android:layout_marginTop="16dp"
                android:layout_weight="1"
                android:background="@drawable/textbox"
                android:ems="10"
                android:hint="그룹명"
                android:inputType="textPersonName" />

            <EditText
                android:id="@+id/makegroup_password"
                android:layout_width="match_parent"
                android:layout_height="match_parent"
                android:layout_marginBottom="12dp"
                android:layout_marginLeft="16dp"
                android:layout_marginRight="16dp"
                android:layout_marginTop="12dp"
                android:layout_weight="1"
                android:background="@drawable/textbox"
                android:ems="10"
                android:hint="비밀번호 4~8 자리"
                android:inputType="numberPassword"
                android:maxLength="8" />

            <LinearLayout
                android:layout_width="match_parent"
                android:layout_height="match_parent"
                android:layout_weight="1"
                android:orientation="horizontal"
                android:padding="0dp">

```



```

<Button
    android:id="@+id/cancel_group"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_marginBottom="16dp"
    android:layout_marginLeft="16dp"
    android:layout_marginRight="8dp"
    android:layout_marginTop="8dp"
    android:layout_weight="2"
    android:background="@drawable/roundbutton_white"
    android:text="닫기"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintHorizontal_bias="1.0"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent" />

<Button
    android:id="@+id/makegroup"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_marginBottom="16dp"
    android:layout_marginLeft="8dp"
    android:layout_marginRight="16dp"
    android:layout_marginTop="8dp"
    android:layout_weight="2"
    android:background="@drawable/roundbutton_white"
    android:text="생성"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintHorizontal_bias="1.0"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent" />
</LinearLayout>

</LinearLayout>

<LinearLayout
    android:id="@+id/tab2_group"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">

    <EditText
        android:id="@+id/joingroup_name"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_marginBottom="8dp"
        android:layout_marginLeft="16dp"
        android:layout_marginRight="16dp"
        android:layout_marginTop="16dp"
        android:layout_weight="1"
        android:background="@drawable/textbox"
        android:ems="10"
        android:hint="그룹명"
        android:inputType="textPersonName"
        tools:layout_editor_absoluteX="8dp"
        tools:layout_editor_absoluteY="8dp" />

    <EditText
        android:id="@+id/joingroup_password"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_marginBottom="12dp"

```

```

        android:layout_marginLeft="16dp"
        android:layout_marginRight="16dp"
        android:layout_marginTop="12dp"
        android:layout_weight="1"
        android:background="@drawable/textbox"
        android:ems="10"
        android:hint="비밀번호 4~8 자리"
        android:inputType="numberPassword"
        android:maxLength="8"
        tools:layout_editor_absoluteX="8dp"
        tools:layout_editor_absoluteY="8dp" />

```

```
<LinearLayout
```

```

        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_weight="1"
        android:orientation="horizontal"
        android:padding="0dp"
        tools:layout_editor_absoluteX="8dp"
        tools:layout_editor_absoluteY="8dp">

```

```
<Button
```

```

        android:id="@+id/cancel_joiningroup"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_marginBottom="16dp"
        android:layout_marginLeft="16dp"
        android:layout_marginRight="8dp"
        android:layout_marginTop="8dp"
        android:layout_weight="2"
        android:background="@drawable/roundbutton_white"
        android:text="닫기"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintHorizontal_bias="1.0"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent" />

```

```
<Button
```

```

        android:id="@+id/joiningroup"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_marginBottom="16dp"
        android:layout_marginLeft="8dp"
        android:layout_marginRight="16dp"
        android:layout_marginTop="8dp"
        android:layout_weight="2"
        android:background="@drawable/roundbutton_white"
        android:text="가입"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintHorizontal_bias="1.0"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent" />

```

```
</LinearLayout>
```

```
</LinearLayout>
```

```
</FrameLayout>
```

```
</LinearLayout>
```

```
</TabHost>
```

```
</LinearLayout>
```

■ 기능 구현

GrouppopupActivity.xml

```

package com.example.kimhyju.place;

import android.app.Activity;
import android.content.Intent;
import android.os.AsyncTask;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.view.Window;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TabHost;
import android.widget.Toast;

import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

import java.io.BufferedReader;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.net.HttpURLConnection;
import java.net.MalformedURLException;
import java.net.URL;
import java.util.ArrayList;
import java.util.UUID;

public class GrouppopupActivity extends Activity implements View.OnClickListener{
    TabHost tabHost;
    EditText makegroup_name, joingroup_name;
    EditText makegroup_password, joingroup_password;
    String groupname;
    String grouppassword;
    Button cancel_group, makegroup, cancel_joingroup, joingroup;
    String userID;
    private static String TAG="phptest_listActivity";
    private static final String TAG_JSON="kimhyju";
    private static final String TAG_PASSWORD="group_password";
    String mJsonString;
    String address;
    String pw, gn;
    ArrayList<String> list;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        requestWindowFeature(Window.FEATURE_NO_TITLE);
        setContentView(R.layout.activity_grouppopup);
        NetworkUtil.setNetworkPolicy();
        makegroup_name=(EditText)findViewById(R.id.makegroup_name);
        makegroup_password=(EditText)findViewById(R.id.makegroup_password);
        cancel_group=(Button)findViewById(R.id.cancel_group);
        makegroup=(Button)findViewById(R.id.makegroup);
        tabHost=(TabHost)findViewById(R.id.tab_host_group);
        cancel_joingroup=(Button)findViewById(R.id.cancel_joingroup);
        joingroup=(Button)findViewById(R.id.joingroup);
        joingroup_name=(EditText)findViewById(R.id.joingroup_name);
        joingroup_password=(EditText)findViewById(R.id.joingroup_password);
        tabHost.setup();
        TabHost.TabSpec tab1 = tabHost.newTabSpec("1").setContent(R.id.tab1_group).setIndicator("그룹
        생성하기");
        TabHost.TabSpec tab2 = tabHost.newTabSpec("2").setContent(R.id.tab2_group).setIndicator("그룹

```

```

가입하기");
    tabHost.addTab(tab1);
    tabHost.addTab(tab2);
    cancel_group.setOnClickListener(this);
    makegroup.setOnClickListener(this);
    joingroup.setOnClickListener(this);
    cancel_joingroup.setOnClickListener(this);
    userID=getIntent().getStringExtra("userid");
    list=getIntent().getStringArrayListExtra("list");

}

//그룹 생성
public boolean addgroup(){
    boolean flag=true;
    try{
        PHPRequest request=new PHPRequest("http://180.71.13.212:8181/addgroup.php");
        String
        result=request.PhPaddgroup(String.valueOf(groupname),String.valueOf(grouppassword));
        if(result.equals("1")){
            flag=true;
            Toast.makeText(getApplicationContext(),"그룹이
            생성되었습니다!",Toast.LENGTH_LONG).show();
        }else if(result.equals("-1")){
            //그룹명을 Primary Key 로 지정해 그룹명이 중복되면 그룹생성이 안되도록한다.
            Toast.makeText(getApplicationContext(),"이미 사용되고 있는
            그룹명입니다!",Toast.LENGTH_LONG).show();
            flag=false;
        }
    }catch (MalformedURLException e){
        e.printStackTrace();
    }
    return flag;
}

//그룹을 생성하면 그룹에 나를 포함시킨다.
public void adduserintogroup(){
    try{
        PHPRequest request=new PHPRequest("http://180.71.13.212:8181/adduserintogroup.php");
        String relationid=UUID.randomUUID().toString();
        String
        result=request.PhPadduserintogroup(String.valueOf(relationid),String.valueOf(groupname),String.valueOf
        (userID));
        if(result.equals("1")){
        }else if(result.equals("-1")){
        }
    }catch (MalformedURLException e){
        e.printStackTrace();
    }
}

//그룹에 가입
public void joingroup(){
    try{
        PHPRequest request=new PHPRequest("http://180.71.13.212:8181/adduserintogroup.php");
        String relationid=UUID.randomUUID().toString();
        String
        result=request.PhPadduserintogroup(String.valueOf(relationid),String.valueOf(gn),String.valueOf(userID
        ));
        if(result.equals("1")){
            Toast.makeText(getApplicationContext(),"그룹에
            가입했습니다!",Toast.LENGTH_LONG).show();
        }
    }
}

```

```

    }else if(result.equals("-1")){

    }
} catch (MalformedURLException e){
    e.printStackTrace();
}
}

@Override
public void onClick(View v) {
    switch(v.getId()){
        case R.id.cancel_group:
            finish();
            break;
        case R.id.makegroup:
            groupname=makegroup_name.getText().toString();
            grouppassword=makegroup_password.getText().toString();
            boolean flag=addgroup();
            if(flag==true)adduser intogroup();
            Intent intent=new Intent(GrouppopupActivity.this,GroupLoadingActivity.class);
            intent.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
            intent.putExtra("userid",user ID);
            startActivity(intent);
            finish();
            break;
        case R.id.cancel_joiningroup:
            finish();
            break;
        case R.id.joiningroup:
            gn=joiningroup_name.getText().toString();
            pw=joiningroup_password.getText().toString();
            //현재 리스트에 없는 그룹에만 가입할수 있도록함.
            if(!list.contains(gn)) {
                address = "http://180.71.13.212:8181/getgroup_password.php?group_name=" + gn;
                //비밀번호가 일치해야지만 가입되도록 하는 코딩을 GetData 클래스 안에서 구현
                GetData task = new GetData();
                task.execute(address);
            }
            else{
                Toast.makeText(getApplicationContext(),"이미 그룹에
가입되어있습니다!",Toast.LENGTH_LONG).show();
                finish();
            }

            break;
    }
}

private class GetData extends AsyncTask<String, Void, String> {
    String errorString = null;
    @Override
    protected void onPreExecute() {
        super.onPreExecute();
    }
    @Override
    protected void onPostExecute(String result) {
        super.onPostExecute(result);
        mJsonString=result;
        try {
            showResult();
        } catch (Exception e){
            Log.d("error", "에러발생");
        }
    }
}

```

```

        Intent intent=new Intent(GrouppopupActivity.this,GroupLoadingActivity.class);
        intent.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
        intent.putExtra("user id",user ID);
        startActivity(intent);
        finish();
    }

    @Override
    protected String doInBackground(String... params) {
        String serverURL = params[0];
        try {
            URL url = new URL(serverURL);
            HttpURLConnection httpURLConnection = (HttpURLConnection) url.openConnection();
            httpURLConnection.setReadTimeout(5000);
            httpURLConnection.setConnectTimeout(5000);
            httpURLConnection.connect();
            int responseStatusCode = httpURLConnection.getResponseCode();
            Log.d(TAG, "response code - " + responseStatusCode);
            InputStream inputStream;
            if(responseStatusCode == HttpURLConnection.HTTP_OK) {
                inputStream = httpURLConnection.getInputStream();
            }
            else{
                inputStream = httpURLConnection.getErrorStream();
            }
            InputStreamReader inputStreamReader = new InputStreamReader(inputStream, "UTF-8");
            BufferedReader bufferedReader = new BufferedReader(inputStreamReader);
            StringBuilder sb = new StringBuilder();
            String line;
            while((line = bufferedReader.readLine()) != null){
                sb.append(line);
            }
            bufferedReader.close();
            return sb.toString().trim();
        } catch (Exception e) {
            Log.d(TAG, "InsertData: Error ", e);
            errorString = e.toString();
            return null;
        }
    }
}

private void showResult(){
    try {
        JSONObject jsonObject = new JSONObject(mJsonString);
        JSONArray jsonArray = jsonObject.getJSONArray( TAG_JSON );
        for( int i=0; i<javascriptArray.length(); i++){
            JSONObject item = jsonArray.getJSONObject(i);
            String password = item.getString( TAG_PASSWORD );
            //비밀번호가 일치하면 그룹에 가입하는 메소드인 joingroup 을 실행
            if(pw.equals(password)){
                joingroup();
            }
            else{
                Toast.makeText(getApplicationContext(),"비밀번호가
틀렸습니다!",Toast.LENGTH_LONG).show();
            }
        }
    } catch (JSONException e) {
        Log.d(TAG, "showResult : ", e);
    }
}
}
}

```

8) 친구목록로딩 액티비티

■ 레이아웃 구현

화면(UI)	설명
화면없음	<ul style="list-style-type: none"> - 메인화면은 타이틀과 리스트뷰 그리고 New Memo, Close 버튼으로 구성 - 저장된 메모를 보여주는 리스트뷰는 어찌구...저찌구 <p><i>레이아웃 설명하세요.</i></p>
레이아웃 없음	
레이아웃없음	

■ 기능 구현

FriendLoadingActivity.java
<pre> package com.example.kimhyju.place; import android.app.Activity; import android.content.Intent; import android.os.AsyncTask; import android.os.Bundle; import android.support.annotation.Nullable; import android.util.Log; import org.json.JSONArray; import org.json.JSONException; import org.json.JSONObject; import java.io.BufferedReader; import java.io.InputStream; import java.io.InputStreamReader; import java.net.HttpURLConnection; import java.net.URL; import java.util.ArrayList; public class FriendLoadingActivity extends Activity{ private static String TAG="phptest_listActivity"; private static final String TAG_JSON="kimhyju"; ArrayList<String> list; String mJsonString; String userId; String address; ArrayList<FriendItem> friendItems; @Override protected void onCreate(@Nullable Bundle savedInstanceState) { super.onCreate(savedInstanceState); NetworkUtil.setNetworkPolicy(); list=new ArrayList<>(); userId=getIntent().getStringExtra("user id"); address="http://180.71.13.212:8181/getfriend.php?me_user id="+userId; friendItems=new ArrayList<>(); GetData task=new GetData(); task.execute(address); } </pre>

//친구목록을 데이터베이스에서 받아오고 ArrayList 에 저장한다. 친구목록을 모두 받아오면 intent 에 리스트를 넘겨주고 FriendActivity 를 실행한다.

```
private class GetData extends AsyncTask<String, Void, String> {
    String errorString = null;
    @Override
    protected void onPreExecute() {
        super.onPreExecute();
    }
    @Override
    protected void onPostExecute(String result) {
        mJsonString=result;
        try {
            showResult();
        }catch(Exception e){

        }
        Intent intent=new Intent(FriendLoadingActivity.this,FriendActivity.class);
        intent.putExtra("friendlist",friendItems);
        intent.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
        startActivity(intent);
        finish();
        super.onPostExecute(result);
    }
    @Override
    protected String doInBackground(String... params) {
        String serverURL = params[0];
        try {
            URL url = new URL(serverURL);
            HttpURLConnection httpURLConnection = (HttpURLConnection) url.openConnection();
            httpURLConnection.setReadTimeout(5000);
            httpURLConnection.setConnectTimeout(5000);
            httpURLConnection.connect();
            int responseStatusCode = httpURLConnection.getResponseCode();
            Log.d(TAG, "response code - " + responseStatusCode);
            InputStream inputStream;
            if(responseStatusCode == HttpURLConnection.HTTP_OK) {
                inputStream = httpURLConnection.getInputStream();
            }
            else{
                inputStream = httpURLConnection.getErrorStream();
            }
            InputStreamReader inputStreamReader = new InputStreamReader(inputStream, "UTF-8");
            BufferedReader bufferedReader = new BufferedReader(inputStreamReader);
            StringBuilder sb = new StringBuilder();
            String line;
            while((line = bufferedReader.readLine()) != null){
                sb.append(line);
            }
            bufferedReader.close();
            return sb.toString().trim();
        } catch (Exception e) {
            Log.d(TAG, "InsertData: Error ", e);
            errorString = e.toString();
            return null;
        }
    }
}

private void showResult(){
    try {
        JSONObject jsonObject = new JSONObject(mJsonString);
        JSONArray jsonArray = jsonObject.getJSONArray( TAG_JSON );
        for( int i=0; i<jarray.length(); i++){
```



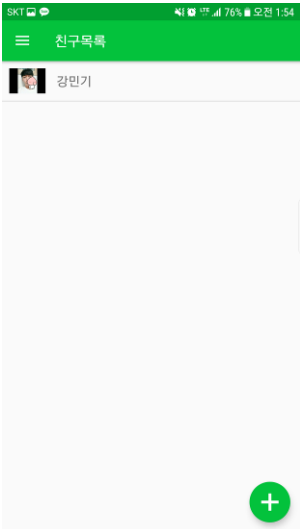
```

        JSONObject item = jsonArray.getJSONObject(i);
        String friendname = item.getString("friend_name");
        String frienduserid=item.getString("friend_userid");
        String friendimage = item.getString("friend_image");
        FriendItem friendItem=new FriendItem(friendimage,friendname,frienduserid);
        friendItems.add(friendItem);
    }
} catch (JSONException e) {
    Log.d(TAG, "showResult : ", e);
}
}
}

```

9) 친구목록화면

■ 레이아웃 구현

화면(UI)	설명
	<ul style="list-style-type: none"> - 메뉴버튼, 플로팅버튼, 리스트로 이루어져있다. -플로팅버튼을 클릭하면 친구추가하기 액티비티가 실행된다. -리스트뷰는 레이아웃을 직접 제작하여 어댑터로 구현하였다.
activity_friend.xml	
<pre> <?xml version="1.0" encoding="utf-8" ?> <android.support.v4.widget.DrawerLayout xmlns:android="http://schemas.android.com/apk/res/android" xmlns:app="http://schemas.android.com/apk/res-auto" xmlns:tools="http://schemas.android.com/tools" android:id="@+id/drawer_layout" android:layout_width="match_parent" android:layout_height="match_parent" android:fitsSystemWindows="true" tools:openDrawer="start"> <include layout="@layout/app_bar_friend" android:layout_width="match_parent" android:layout_height="match_parent" /> <android.support.design.widget.NavigationView android:id="@+id/nav_view" android:layout_width="wrap_content" android:layout_height="match_parent" </pre>	

```

        android:layout_gravity="start"
        android:fitsSystemWindows="true"
        app:headerLayout="@layout/nav_header_main2"
        app:menu="@menu/activity_main2_drawer" />

```

```
</android.support.v4.widget.DrawerLayout>
```

app_bar_friend.xml

```

<?xml version="1.0" encoding="utf-8"?>
<android.support.design.widget.CoordinatorLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="com.example.kimhyju.place.FriendActivity">

    <android.support.design.widget.AppBarLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:theme="@style/AppTheme.AppBarOverlay">

        <android.support.v7.widget.Toolbar
            android:id="@+id/toolbar_friend"
            android:layout_width="match_parent"
            android:layout_height="?attr/actionBarSize"
            android:background="?attr/colorPrimary"
            app:popupTheme="@style/AppTheme.PopupOverlay" />

    </android.support.design.widget.AppBarLayout>

    <include layout="@layout/content_friend" />

    <android.support.design.widget.FloatingActionButton
        android:id="@+id/fab"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="bottom|end"
        android:layout_margin="@dimen/fab_margin"
        app:backgroundTint="?attr/colorPrimary"
        app:srcCompat="@drawable/add" />

</android.support.design.widget.CoordinatorLayout>

```

content_friend.xml

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    app:layout_behavior="@string/appbar_scrolling_view_behavior"
    tools:context="com.example.kimhyju.place.FriendActivity">

    <ListView
        android:id="@+id/friend_list"
        android:layout_width="match_parent"
        android:layout_height="match_parent" />

</LinearLayout>

```

■ 기능 구현

FriendActivity.java

```

package com.example.kimhyju.place;

import android.content.DialogInterface;
import android.content.Intent;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.os.Bundle;
import android.os.Handler;
import android.support.design.widget.FloatingActionButton;
import android.support.design.widget.NavigationView;
import android.support.v4.view.GravityCompat;
import android.support.v4.widget.DrawerLayout;
import android.support.v7.app.ActionBarDrawerToggle;
import android.support.v7.app.AlertDialog;
import android.support.v7.app.AppCompatActivity;
import android.support.v7.widget.PopupMenu;
import android.support.v7.widget.Toolbar;
import android.util.Log;
import android.view.MenuItem;
import android.view.View;
import android.widget.AdapterView;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.ListView;
import android.widget.TextView;
import android.widget.Toast;

import com.kakao.network.ErrorResult;
import com.kakao.usermgmt.UserManagement;
import com.kakao.usermgmt.callback.LogoutResponseCallback;
import com.kakao.usermgmt.callback.MeResponseCallback;
import com.kakao.usermgmt.callback.UnLinkResponseCallback;
import com.kakao.usermgmt.response.model.UserProfile;
import com.kakao.util.helper.log.Logger;

import java.io.InputStream;
import java.net.MalformedURLException;
import java.net.URL;
import java.util.ArrayList;

public class FriendActivity extends AppCompatActivity
    implements NavigationView.OnNavigationItemSelectedListener, View.OnClickListener{
    private static String TAG = "phptest_MainActivity";
    View nav_head_view;
    private ImageView profileImage;
    private TextView nameText;
    private Button confirmID;
    String plImage="";
    String nickName = "";
    long userID = 0;
    String userId;
    String user_email;
    TextView emailText;
    Toolbar toolbar;
    ListView friend_list;
    ArrayList<FriendItem> friends;
    FriendAdapter adapter;
    String friendid1;
    String friendid2;
    @Override
    protected void onCreate(Bundle savedInstanceState) {

```

```

super.onCreate(savedInstanceState);
setContentView(R.layout.activity_friend);
NetworkUtil.setNetworkPolicy();
requestMe();
NavigationView navigationView = (NavigationView) findViewById(R.id.nav_view);
nav_head_view=navigationView.getHeaderView(0);
profileImage=(ImageView)nav_head_view.findViewById(R.id.profileImage);
nameText=(TextView)nav_head_view.findViewById(R.id.nameText1);
emailText=(TextView)nav_head_view.findViewById(R.id.email);
confirmID=(Button)nav_head_view.findViewById(R.id.confirmID);
toolbar = (Toolbar) findViewById(R.id.toolbar_friend);
friend_list=(ListView) findViewById(R.id.friend_list);
setSupportActionBar(toolbar);
getSupportActionBar().setTitle("친구목록");
friends=(ArrayList<FriendItem>) getIntent().getSerializableExtra("friendlist");

//친구 추가하기 FloatingActionButton
FloatingActionButton fab = (FloatingActionButton) findViewById(R.id.fab);
fab.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        Intent intent=new Intent(FriendActivity.this,FriendpopupActivity.class);
        intent.putExtra("user id",userId);
        intent.putExtra("username",nickName);
        intent.putExtra("user image",pImage);
        intent.putExtra("friendlist",friends);
        startActivity(intent);
    }
});

DrawerLayout drawer = (DrawerLayout) findViewById(R.id.drawer_layout);
ActionBarDrawerToggle toggle = new ActionBarDrawerToggle(
    this, drawer, toolbar, R.string.navigation_drawer_open,
R.string.navigation_drawer_close);
drawer.setDrawerListener(toggle);
toggle.syncState();
navigationView.setItemSelectedListener(this);
confirmID.setOnClickListener(this);

//친구목록을 길게 터치하면 삭제버튼이 나타난다.
friend_list.setOnItemLongClickListener(new AdapterView.OnItemLongClickListener() {
    @Override
    public boolean onItemLongClick(AdapterView<?> parent, View view, int position, long id) {
        PopupMenu popup=new PopupMenu(FriendActivity.this,view);
        getMenuInflater().inflate(R.menu.popupmenu,popup.getMenu());
        final int index=position;
        popup.setOnMenuItemClickListener(new PopupMenu.OnMenuItemClickListener() {
            @Override
            public boolean onMenuItemClick(MenuItem item) {
                if(item.getItemId()==R.id.action_delete){
                    String frienduserId=friends.get(index).getUserid();
                    friendid1=userId+frienduserId;
                    friendid2=frienduserId+userId;
                    deletefriend1();
                    deletefriend2();
                    Intent intent=new Intent(FriendActivity.this,FriendLoadingActivity.class);
                    intent.putExtra("user id",userId);
                    intent.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
                    startActivity(intent);
                    finish();
                }
                return false;
            }
        });
    }
});

```

```

        });
        popup.show();
        return false;
    }
});
adapter=new FriendAdapter(getApplicationContext(),R.layout.friendlist,friends);
friend_list.setAdapter(adapter);
}

@Override
public void onBackPressed() {
    DrawerLayout drawer = (DrawerLayout) findViewById(R.id.drawer_layout);
    if (drawer.isDrawerOpen(GravityCompat.START)) {
        drawer.closeDrawer(GravityCompat.START);
    } else {
        finish();
    }
}

@SuppressWarnings("StatementWithEmptyBody")
@Override
public boolean onNavigationItemSelected(MenuItem item) {
    // Handle navigation view item clicks here.
    int id = item.getItemId();
    item.setCheckable(false);
    if (id == R.id.friend_list) {
        // Handle the camera action
    } else if (id == R.id.group_list) {
        Intent intent=new Intent(FriendActivity.this, GroupLoadingActivity.class);
        intent.putExtra("userid",userid);
        startActivity(intent);
        finish();
    } else if (id == R.id.logout) {
        UserManagement.requestLogout(new LogoutResponseCallback() {
            @Override
            public void onCompleteLogout() {
                Intent intent = new Intent(FriendActivity.this, LoginActivity.class);
                startActivity(intent);
                overridePendingTransition(0,0);
                finish();
            }
        });
    } else if (id == R.id.unlink) {
        onClickUnlink();

    }else if(id==R.id.main){
        finish();
    }
    DrawerLayout drawer = (DrawerLayout) findViewById(R.id.drawer_layout);
    drawer.closeDrawer(GravityCompat.START);
    return true;
}

Handler handler = new Handler();
public void LinkImage() {
    if (plmage.equals("img")) ;
    else {
        //new ImageDownload().execute(plmage);
        // 인터넷 상의 이미지 보여주기

        // 1. 권한을 획득한다 (인터넷에 접근할수 있는 권한을 획득한다) - 매니페스트 파일
        // 2. Thread 에서 웹의 이미지를 가져오기
    }
}

```

// 3. 외부쓰레드에서 메인 UI 에 접근위해 Handler 사용

//Thread t = new Thread(Runnable 객체 생성);

Thread t = new Thread(new Runnable() {

@Override

public void run() { // 오래 걸릴 작업 구현

// TODO Auto-generated method stub

try {

final URL url = new URL(pImage);

InputStream is = url.openStream();

final Bitmap bm = BitmapFactory.decodeStream(is);

handler.post(new Runnable() {

@Override

public void run() { // 화면에 그려줄 작업

profileImage.setImageBitmap(bm);

Log.d("imageurl",String.valueOf(url));

}

});

profileImage.setImageBitmap(bm); //비트맵 객체로 보여주기

} catch (Exception e) {

}

}

});

t.start();

}

}

private void onClickUnlink() {

final String appendMessage = getString(R.string.com_kakao_confirm_unlink);

new AlertDialog.Builder(this)

.setMessage(appendMessage)

.setPositiveButton(getString(R.string.com_kakao_ok_button),

new DialogInterface.OnClickListener() {

@Override

public void onClick(DialogInterface dialog, int which) {

UserManagement.requestUnlink(new UnLinkResponseCallback() {

@Override

public void onFailure(ErrorResult errorResult) {

Logger.e(errorResult.toString());

}

@Override

public void onSessionClosed(ErrorResult errorResult) {

//redirectLoginActivity();

}

@Override

public void onNotSignedUp() {

//redirectSignupActivity();

}

@Override

public void onSuccess(Long userId) {

deleteuser(userID);

Toast.makeText(getApplicationContext(), "탈퇴되었습니다.",

Toast.LENGTH_LONG).show();

//redirectLoginActivity();

Intent intent = new Intent(getApplicationContext(),

LoginActivity.class);

startActivity(intent);

overridePendingTransition(0,0);

finish();

}

});

dialog.dismiss();

}

```

        })
        .setNegativeButton(getString(R.string.com_kakao_cancel_button),
            new DialogInterface.OnClickListener() {
                @Override
                public void onClick(DialogInterface dialog, int which) {
                    dialog.dismiss();
                }
            }).show();
    }

    @Override
    public void onClick(View v) {
        if(v.getId()==R.id.confirmID){
            Toast.makeText(getApplicationContext(),"당신의 ID 번호는
"+user ID+"입니다.",Toast.LENGTH_LONG).show();
        }
    }

    private void requestMe() {
        UserManagement.requestMe(new MeResponseCallback() {
            @Override
            public void onFailure(ErrorResult errorResult) {
                String message = "failed to get user info. msg=" + errorResult;
                Logger.d(message);
                //redirectLoginActivity();
            }
            @Override
            public void onSessionClosed(ErrorResult errorResult) {
                //redirectLoginActivity();
            }
            @Override
            public void onSuccess(UserProfile userProfile) {
                Logger.d("UserProfile : " + userProfile);
                nickname = userProfile.getNickname();//닉네임
                userID = userProfile.getId();//사용자 고유번호
                plmage = userProfile.getProfileImagePath();//사용자 프로필 경로
                user_email=userProfile.getEmail();
                emailText.setText(user_email);
                nameText.setText(nickname);
                user id=String.valueOf(userID);
                LinkImage();
            }
            @Override
            public void onNotSignedUp() {
                //showSignup();
            }
        });
    }

    public void deleteuser(long userID){
        try{
            PHPRequest request=new PHPRequest("http://180.71.13.212:8181/delete.php");
            String result=request.PhDelete(String.valueOf(userID));
            if(result.equals("1")){
                Toast.makeText(getApplicationContext(),"Success!",Toast.LENGTH_LONG).show();
            }else if(result.equals("-1")){
                Toast.makeText(getApplicationContext(),"Fail!",Toast.LENGTH_LONG).show();
            }
        }catch (MalformedURLException e){
            e.printStackTrace();
        }
    }
}

```

```

//친구 삭제(나-친구)
public void deletefriend1(){
    try{
        PHPRequest request=new PHPRequest("http://180.71.13.212:8181/deletefriend.php");
        String result=request.PhDeletefriend(String.valueOf(friendid1));
        if(result.equals("1")){
            Toast.makeText(getApplicationContext(),"친구를
삭제했습니다!",Toast.LENGTH_LONG).show();
        }else if(result.equals("-1")){
        }
    }catch (MalformedURLException e){
        e.printStackTrace();
    }
}

//친구삭제(친구-나)
public void deletefriend2(){
    try{
        PHPRequest request=new PHPRequest("http://180.71.13.212:8181/deletefriend.php");
        String result=request.PhDeletefriend(String.valueOf(friendid2));
        if(result.equals("1")){
        }else if(result.equals("-1")){
        }
    }catch (MalformedURLException e){
        e.printStackTrace();
    }
}
}
}

```

10) 친구 추가하기 화면

■ 레이아웃 구현

화면(UI)	설명
	<p>- 추가할친구의 스마트폰에서 어플을 실행시킨뒤 메뉴를 열고 내ID확인 버튼을 클릭하면 Toast메시지로 나타나는 고유 ID번호를 나의 앱에 입력하면 친구가 추가된다.</p>
activity_friendpopup.xml	
<pre> <?xml version="1.0" encoding="utf-8" ?> <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android" xmlns:app="http://schemas.android.com/apk/res-auto" xmlns:tools="http://schemas.android.com/tools" android:layout_width="310dp" android:layout_height="210dp" android:orientation="vertical" tools:context="com.example.kimhyju.place.FriendpopupActivity"> </pre>	


```
<TextView
    android:id="@+id/textView3"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_marginBottom="8dp"
    android:layout_marginLeft="16dp"
    android:layout_marginRight="16dp"
    android:layout_marginTop="16dp"
    android:layout_weight="1"
    android:gravity="center_vertical|center_horizontal"
    android:text="친구의 ID 번호를 입력하세요."
    android:textSize="20sp" />

<EditText
    android:id="@+id/friendID"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_marginBottom="12dp"
    android:layout_marginLeft="16dp"
    android:layout_marginRight="16dp"
    android:layout_marginTop="12dp"
    android:layout_weight="1"
    android:background="@drawable/textbox"
    android:ems="10"
    android:hint="친구의 ID 번호"
    android:inputType="numberPassword"
    android:maxLength="10" />

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_weight="1"
    android:orientation="horizontal"
    android:padding="0dp">

    <Button
        android:id="@+id/cancel_friend"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_marginBottom="16dp"
        android:layout_marginLeft="16dp"
        android:layout_marginRight="8dp"
        android:layout_marginTop="8dp"
        android:layout_weight="2"
        android:background="@drawable/roundbutton_white"
        android:text="닫 기"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintHorizontal_bias="1.0"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent" />

    <Button
        android:id="@+id/addfriend"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_marginBottom="16dp"
        android:layout_marginLeft="8dp"
        android:layout_marginRight="16dp"
        android:layout_marginTop="8dp"
        android:layout_weight="2"
        android:background="@drawable/roundbutton_white"
        android:text="추 가"
        app:layout_constraintBottom_toBottomOf="parent"
```

```

        app:layout_constraintHorizontal_bias="1.0"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent" />
    </LinearLayout>
</LinearLayout>

```

■ 기능 구현

FriendpopupActivity.xml

```

package com.example.kimhyju.place;

import android.app.Activity;
import android.content.Intent;
import android.os.AsyncTask;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.view.Window;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

import java.io.BufferedReader;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.net.HttpURLConnection;
import java.net.MalformedURLException;
import java.net.URL;
import java.util.ArrayList;

public class FriendpopupActivity extends Activity implements View.OnClickListener{
    String friend_userid, friend_name="", friend_image="";
    EditText friend_userid_input;
    Button cancel_friend, addfriend;
    String me_userid, me_name, me_image;
    String friendid1, friendid2;
    String mJsonString;
    String address;
    ArrayList<FriendItem> friendlist;
    Boolean iscontained;
    private static String TAG="phptest_listActivity";
    private static final String TAG_JSON="kimhyju";
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        requestWindowFeature(Window.FEATURE_NO_TITLE);
        setContentView(R.layout.activity_friendpopup);
        NetworkUtil.setNetworkPolicy();
        me_userid=getIntent().getStringExtra("user id");
        me_name=getIntent().getStringExtra("user name");
        me_image=getIntent().getStringExtra("user image");
        friend_userid_input=(EditText)findViewById(R.id.friendID);
        cancel_friend=(Button)findViewById(R.id.cancel_friend);
        addfriend=(Button)findViewById(R.id.addfriend);
        cancel_friend.setOnClickListener(this);
        addfriend.setOnClickListener(this);
    }

```

```

        friendlist=new ArrayList<>();
        friendlist=(ArrayList<FriendItem>)getIntent().getSerializableExtra("friendlist");
    }

    @Override
    public void onClick(View v) {
        switch(v.getId()){
            case R.id.cancel_friend:
                finish();
                break;
            case R.id.addfriend:
                //추가하려는 친구가 이미 친구로 존재하지 않을때만 친구가 추가되도록함
                friend_userid=friend_userid_input.getText().toString();
                for(int i=0; i<friendlist.size();i++){
                    if(friendlist.get(i).getUserid().equals(friend_userid)){
                        iscontained=true;
                        Toast.makeText(getApplicationContext(),"이미
친구입니다!",Toast.LENGTH_LONG).show();
                        break;
                    }else{
                        iscontained=false;
                    }
                }
                if(iscontained==false) {
                    friendid1 = me_userid + friend_userid;
                    friendid2 = friend_userid + me_userid;
                    address = "http://180.71.13.212:8181/getfriendinfo.php?userid=" + friend_userid;
                    GetData task = new GetData();
                    task.execute(address);
                }
                break;
        }
    }

    //친구추가(나-친구)
    public void addfriend1(){
        try{
            PHPRequest request=new PHPRequest("http://180.71.13.212:8181/addfriend.php");
            String
            result=request.PhPaddfriend(String.valueOf(friendid1),String.valueOf(me_name),String.valueOf(me_userid
            ),String.valueOf(friend_name),String.valueOf(friend_userid),String.valueOf(friend_image));
            if(result.equals("1")){
                Toast.makeText(getApplicationContext(),"친구가
추가되었습니다!",Toast.LENGTH_LONG).show();
            }else if(result.equals("-1")){

            }
        }catch (MalformedURLException e){
            e.printStackTrace();
        }
    }

    //친구추가 (친구-나)
    public void addfriend2(){
        try{
            PHPRequest request=new PHPRequest("http://180.71.13.212:8181/addfriend.php");
            String
            result=request.PhPaddfriend(String.valueOf(friendid2),String.valueOf(friend_name),String.valueOf(frien
            d_userid),String.valueOf(me_name),String.valueOf(me_userid),String.valueOf(me_image));
            if(result.equals("1")){
            }else if(result.equals("-1")){
            }
        }catch (MalformedURLException e){

```

```

        e.printStackTrace();
    }
}

private class GetData extends AsyncTask<String, Void, String> {
    String errorString = null;
    @Override
    protected void onPreExecute() {
        super.onPreExecute();
    }
    @Override
    protected void onPostExecute(String result) {
        mJsonString=result;
        try {
            showResult();
        } catch (Exception e) {
        }
        if (!friend_name.equals("") && !friend_image.equals("")) {
            addfriend1();
            addfriend2();
        } else {
            Toast.makeText(getApplicationContext(), "없는 친구입니다!", Toast.LENGTH_LONG).show();
        }
        Intent intent=new Intent(FriendpopupActivity.this, FriendLoadingActivity.class);
        intent.putExtra("userid", me_userid);
        intent.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
        startActivity(intent);
        finish();
        super.onPostExecute(result);
    }
}

@Override
protected String doInBackground(String... params) {

    String serverURL = params[0];

    try {
        URL url = new URL(serverURL);
        HttpURLConnection httpURLConnection = (HttpURLConnection) url.openConnection();
        httpURLConnection.setReadTimeout(5000);
        httpURLConnection.setConnectTimeout(5000);
        httpURLConnection.connect();
        int responseStatusCode = httpURLConnection.getResponseCode();
        Log.d(TAG, "response code - " + responseStatusCode);
        InputStream inputStream;
        if (responseStatusCode == HttpURLConnection.HTTP_OK) {
            inputStream = httpURLConnection.getInputStream();
        }
        else {
            inputStream = httpURLConnection.getErrorStream();
        }
        InputStreamReader inputStreamReader = new InputStreamReader(inputStream, "UTF-8");
        BufferedReader bufferedReader = new BufferedReader(inputStreamReader);
        StringBuilder sb = new StringBuilder();
        String line;
        while ((line = bufferedReader.readLine()) != null) {
            sb.append(line);
        }
        bufferedReader.close();
        return sb.toString().trim();
    } catch (Exception e) {
        Log.d(TAG, "InsertData: Error ", e);
    }
}

```

```
        errorString = e.toString();
        return null;
    }
}

private void showResult(){
    try {
        JSONObject jsonObject = new JSONObject(mJsonString);
        JSONArray jsonArray = jsonObject.getJSONArray( TAG_JSON );
        for( int i=0; i<jsonArray.length(); i++){
            JSONObject item = jsonArray.getJSONObject(i);
            String imgurl = item.getString("imgurl");
            String name=item.getString("name");
            friend_image=imgurl;
            friend_name=name;
        }
    } catch (JSONException e) {
        Log.d( TAG, "showResult : ", e);
    }
}
```

11) 그외의 클래스들

■ 그룹 객체

GroupItem.java

```
package com.example.kimhyju.place;

import java.io.Serializable;

public class GroupItem implements Serializable{
    String groupname;
    String relationid;
    public GroupItem(String _relationid,String _groupname){
        this.relationid=_relationid;
        this.groupname=_groupname;
    };
    public GroupItem(String s){
        this.groupname=s;
    }
    public void setGroupname(String s){
        groupname=s;
    }
    public String getGroupname(){
        return groupname;
    }
    public String getRelationid(){return relationid;}
}
```

■ 친구 객체

FriendItem.java

```
package com.example.kimhyju.place;

import java.io.Serializable;

public class FriendItem implements Serializable{
    String plmage;
    String nickName;
    String user id;

    public FriendItem(String _plmage,String _nickName,String _userid){
        plmage=_plmage;
        nickName=_nickName;
        user id=_user id;
    }
    public void setplmage(String _plmage){
        plmage=_plmage;
    }
    public void setNickName(String _nickName){
        nickName=_nickName;
    }
    public String getplmage(){
        return plmage;
    }
    public String getNickName(){
        return nickName;
    }
    public String getUserid(){return user id;}
}
```

■ 마커 객체

MarkerInfo.java

```
package com.example.kimhyju.place;

import java.io.Serializable;

public class MarkerInfo implements Serializable{
    String markerid;
    Double lat;
    Double lng;
    String place_nmae;
    String writer;
    String place_content;
    public MarkerInfo(String _markerid,Double _lat,Double _lng,String _place_name,String
_writer,String _place_content){
        markerid=_markerid;
        lat=_lat;
        lng=_lng;
        place_nmae=_place_name;
        writer=_writer;
        place_content=_place_content;
    }
    public String getMarkerid(){
        return markerid;
    }
    public Double getLat(){
        return lat;
    }
    public Double getLng(){
        return lng;
    }
    public String getPlace_nmae(){
        return place_nmae;
    }
    public String getWriter(){
        return writer;
    }
    public String getPlace_content(){
        return place_content;
    }
    public void setMarkerid(String _markerid){
        markerid=_markerid;
    }
    public void setLat(Double _lat){
        lat=_lat;
    }
    public void setLng(Double _lng){
        lng=_lng;
    }
    public void setPlace_nmae(String _place_name){
        place_nmae=_place_name;
    }
    public void setWriter(String _writer){
        writer=_writer;
    }
    public void setPlace_content(String _place_content){
        place_content=_place_content;
    }
}
```

■ 그룹 ArrayAdapter

GroupAdapter.java

```

package com.example.kimhyju.place;

import android.content.Context;
import android.support.annotation.LayoutRes;
import android.support.annotation.NonNull;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ArrayAdapter;
import android.widget.TextView;

import java.util.ArrayList;

public class GroupAdapter extends ArrayAdapter {
    ArrayList<GroupItem> GroupItemList=new ArrayList<>();
    TextView groupname;

    public GroupAdapter(@NonNull Context context, @LayoutRes int resource, @NonNull
ArrayList<GroupItem> item) {
        super(context, resource, item);
        this.GroupItemList=item;
    }

    @Override
    public View getView(int position, View convertView, ViewGroup parent) {
        final int pos = position;
        final Context context = parent.getContext();
        if (convertView == null) {
            LayoutInflater inflater = (LayoutInflater)
context.getSystemService(Context.LAYOUT_INFLATER_SERVICE);
            convertView = inflater.inflate(R.layout.group_list, parent, false);
        }
        GroupItem groupitem=GroupItemList.get(position);
        groupname=(TextView)convertView.findViewById(R.id.groupname);
        groupname.setText(groupitem.getGroupname());
        return convertView;
    }

    @Override
    public long getItemId(int position) {
        return position;
    }

    @Override
    public Object getItem(int position) {
        return GroupItemList.get(position);
    }

    @Override
    public int getCount() {
        return GroupItemList.size();
    }
}

```


■ 친구 ArrayAdapter

FriendAdapter.java

```

package com.example.kimhyju.place;

import android.content.Context;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.os.Handler;
import android.support.annotation.LayoutRes;
import android.support.annotation.NonNull;
import android.util.Log;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ArrayAdapter;
import android.widget.ImageView;
import android.widget.TextView;

import java.io.InputStream;
import java.net.URL;
import java.util.ArrayList;

public class FriendAdapter extends ArrayAdapter {
    ArrayList<FriendItem> FriendItemList=new ArrayList<>();
    TextView friendname;
    ImageView friendimage;
    public FriendAdapter(@NonNull Context context, @LayoutRes int resource, @NonNull
ArrayList<FriendItem> item) {
        super(context, resource, item);
        this.FriendItemList=item;
    }

    @Override
    public View getView(int position, View convertView, ViewGroup parent) {
        final int pos = position;
        final Context context = parent.getContext();
        if (convertView == null) {
            LayoutInflater inflater = (LayoutInflater)
context.getSystemService(Context.LAYOUT_INFLATER_SERVICE);
            convertView = inflater.inflate(R.layout.friendlist, parent, false);
        }
        FriendItem frienditem=FriendItemList.get(position);
        friendname=(TextView)convertView.findViewById(R.id.friendname);
        friendimage=(ImageView)convertView.findViewById(R.id.friendimage);
        friendname.setText(frienditem.getNickName());
        //친구의 이미지경로를 불러와 이미지뷰에 비트맵으로 입력한다.
        LinkImage(frienditem.getpImage(),friendimage);
        return convertView;
    }

    @Override
    public long getItemId(int position) {
        return position;
    }

    @Override
    public Object getItem(int position) {
        return FriendItemList.get(position);
    }

    @Override
    public int getCount() {
        return FriendItemList.size();
    }

```

```

    }

    Handler handler = new Handler();
    public void LinkImage(final String pImage,final ImageView profileImage) {
        if (pImage.equals("img")) ;
        else {
            Thread t = new Thread(new Runnable() {
                @Override
                public void run() {
                    // TODO Auto-generated method stub
                    try {
                        final URL url = new URL(pImage);
                        InputStream is = url.openStream();
                        final Bitmap bm = BitmapFactory.decodeStream(is);
                        handler.post(new Runnable() {

                            @Override
                            public void run() {
                                profileImage.setImageBitmap(bm);
                                Log.d("imageurl",String.valueOf(url));
                            }
                        });
                        profileImage.setImageBitmap(bm);
                    } catch (Exception e) {
                    }
                }
            });
            t.start();
        }
    }
}

```

■ 네트워크 설정 클래스

NetworkUtil.java

```

package com.example.kimhyju.place;

import android.annotation.SuppressLint;
import android.os.StrictMode;
public class NetworkUtil {
    @SuppressWarnings("NewApi")
    static public void setNetworkPolicy() {
        if (android.os.Build.VERSION.SDK_INT > 9) {
            StrictMode.ThreadPolicy policy = new
StrictMode.ThreadPolicy.Builder().permitAll().build();
            StrictMode.setThreadPolicy(policy);
        }
    }
}

```

■ 데이터베이스에 데이터를 추가하거나 삭제하는 메소드를 포함한 클래스

PHPRequest.java

```

package com.example.kimhyju.place;

import android.util.Log;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.io.OutputStream;
import java.net.HttpURLConnection;
import java.net.MalformedURLException;
import java.net.URL;

//데이터를 입력하거나 삭제할때 선언되는 클래스
public class PHPRequest {
    private URL url;

    public PHPRequest(String url) throws MalformedURLException { this.url = new URL(url); }

    private String readStream(InputStream in) throws IOException {
        StringBuilder jsonHtml = new StringBuilder();
        BufferedReader reader = new BufferedReader(new InputStreamReader(in, "UTF-8"));
        String line = null;

        while((line = reader.readLine()) != null)
            jsonHtml.append(line);

        reader.close();
        return jsonHtml.toString();
    }

    //사용자 등록
    public String PhPinser(final String userid, final String imgurl, final String name) {
        try {
            String postData = "userid=" + userid + "&" + "imgurl=" + imgurl + "&" + "name=" + name;
            HttpURLConnection conn = (HttpURLConnection)url.openConnection();
            conn.setRequestProperty("Content-Type", "application/x-www-form-urlencoded");
            conn.setRequestMethod("POST");
            conn.setConnectTimeout(5000);
            conn.setDoOutput(true);
            conn.setDoInput(true);
            OutputStream outputStream = conn.getOutputStream();
            outputStream.write(postData.getBytes("UTF-8"));
            outputStream.flush();
            outputStream.close();
            String result = readStream(conn.getInputStream());
            conn.disconnect();
            return result;
        }
        catch (Exception e) {
            Log.i("PHPRequest", "request was failed.");
            return null;
        }
    }

    //사용자 삭제
    public String PhPdelete(final String userid) {
        try {
            String postData = "userid=" + userid;

```

```

        HttpURLConnection conn = (HttpURLConnection)url.openConnection();
        conn.setRequestProperty("Content-Type", "application/x-www-form-urlencoded");
        conn.setRequestMethod("POST");
        conn.setConnectTimeout(5000);
        conn.setDoOutput(true);
        conn.setDoInput(true);
        OutputStream outputStream = conn.getOutputStream();
        outputStream.write(postData.getBytes("UTF-8"));
        outputStream.flush();
        outputStream.close();
        String result = readStream(conn.getInputStream());
        conn.disconnect();
        return result;
    }
    catch (Exception e) {
        Log.i("PHPRequest", "request was failed.");
        return null;
    }
}

//친구 삭제
public String PhDeletefriend(final String friendid) {
    try {
        String postData = "friendid=" + friendid;
        HttpURLConnection conn = (HttpURLConnection)url.openConnection();
        conn.setRequestProperty("Content-Type", "application/x-www-form-urlencoded");
        conn.setRequestMethod("POST");
        conn.setConnectTimeout(5000);
        conn.setDoOutput(true);
        conn.setDoInput(true);
        OutputStream outputStream = conn.getOutputStream();
        outputStream.write(postData.getBytes("UTF-8"));
        outputStream.flush();
        outputStream.close();
        String result = readStream(conn.getInputStream());
        conn.disconnect();
        return result;
    }
    catch (Exception e) {
        Log.i("PHPRequest", "request was failed.");
        return null;
    }
}

//그룹삭제
public String PhDeletegroup(final String relationid) {
    try {
        String postData = "relationid=" + relationid;
        HttpURLConnection conn = (HttpURLConnection)url.openConnection();
        conn.setRequestProperty("Content-Type", "application/x-www-form-urlencoded");
        conn.setRequestMethod("POST");
        conn.setConnectTimeout(5000);
        conn.setDoOutput(true);
        conn.setDoInput(true);
        OutputStream outputStream = conn.getOutputStream();
        outputStream.write(postData.getBytes("UTF-8"));
        outputStream.flush();
        outputStream.close();
        String result = readStream(conn.getInputStream());
        conn.disconnect();
        return result;
    }
    catch (Exception e) {

```

```

        Log.i("PHPRequest", "request was failed.");
        return null;
    }
}

```

//마커 삭제

```

public String PhPdeletemarker(final String markerid) {
    try {
        String postData = "markerid=" + markerid;
        HttpURLConnection conn = (HttpURLConnection)url.openConnection();
        conn.setRequestProperty("Content-Type", "application/x-www-form-urlencoded");
        conn.setRequestMethod("POST");
        conn.setConnectTimeout(5000);
        conn.setDoOutput(true);
        conn.setDoInput(true);
        OutputStream outputStream = conn.getOutputStream();
        outputStream.write(postData.getBytes("UTF-8"));
        outputStream.flush();
        outputStream.close();
        String result = readStream(conn.getInputStream());
        conn.disconnect();
        return result;
    }
    catch (Exception e) {
        Log.i("PHPRequest", "request was failed.");
        return null;
    }
}

```

//그룹생성

```

public String PhPaddgroup(final String groupname,final String grouppassword){
    try {
        String postData = "groupname=" + groupname + "&" + "grouppassword=" + grouppassword;
        HttpURLConnection conn = (HttpURLConnection)url.openConnection();
        conn.setRequestProperty("Content-Type", "application/x-www-form-urlencoded");
        conn.setRequestMethod("POST");
        conn.setConnectTimeout(5000);
        conn.setDoOutput(true);
        conn.setDoInput(true);
        OutputStream outputStream = conn.getOutputStream();
        outputStream.write(postData.getBytes("UTF-8"));
        outputStream.flush();
        outputStream.close();
        String result = readStream(conn.getInputStream());
        conn.disconnect();
        return result;
    }
    catch (Exception e) {
        Log.i("PHPRequest", "request was failed.");
        return null;
    }
}

```

//그룹에 사용자 추가

```

public String PhPadduserintogroup(final String relationid, final String groupname,final String
userid){
    try {
        String postData = "relationid=" + relationid + "&" + "groupname=" + groupname + "&" +
"userid=" + userid;
        HttpURLConnection conn = (HttpURLConnection)url.openConnection();
        conn.setRequestProperty("Content-Type", "application/x-www-form-urlencoded");
        conn.setRequestMethod("POST");
        conn.setConnectTimeout(5000);

```

```

        conn.setDoOutput(true);
        conn.setDoInput(true);
        OutputStream outputStream = conn.getOutputStream();
        outputStream.write(postData.getBytes("UTF-8"));
        outputStream.flush();
        outputStream.close();
        String result = readStream(conn.getInputStream());
        conn.disconnect();
        return result;
    }
    catch (Exception e) {
        Log.i("PHPRequest", "request was failed.");
        return null;
    }
}

//마커 생성
public String PhPaddmarker(final String markerid, final String lat, final String lng, final String
name_place, final String name_group, final String content, final String name_user, final String userid){
    try {
        String postData = "markerid=" + markerid + "&" + "lat=" + lat + "&" + "lng=" + lng + "&" +
"name_place=" + name_place + "&" + "name_group=" + name_group + "&" + "content=" + content + "&" +
"name_user=" + name_user + "&" + "userid=" + userid ;
        HttpURLConnection conn = (HttpURLConnection)url.openConnection();
        conn.setRequestProperty("Content-Type", "application/x-www-form-urlencoded");
        conn.setRequestMethod("POST");
        conn.setConnectTimeout(5000);
        conn.setDoOutput(true);
        conn.setDoInput(true);
        OutputStream outputStream = conn.getOutputStream();
        outputStream.write(postData.getBytes("UTF-8"));
        outputStream.flush();
        outputStream.close();
        String result = readStream(conn.getInputStream());
        conn.disconnect();
        return result;
    }
    catch (Exception e) {
        Log.i("PHPRequest", "request was failed.");
        return null;
    }
}

//친구등록
public String PhPaddfriend(final String friendid, final String me_name, final String
me_userid, final String friend_name, final String friend_userid, final String friend_image){
    try {
        String postData = "friendid=" + friendid + "&" + "me_name=" + me_name + "&" + "me_userid="
+ me_userid + "&" + "friend_name=" + friend_name + "&" + "friend_userid=" + friend_userid + "&" +
"friend_image=" + friend_image;
        HttpURLConnection conn = (HttpURLConnection)url.openConnection();
        conn.setRequestProperty("Content-Type", "application/x-www-form-urlencoded");
        conn.setRequestMethod("POST");
        conn.setConnectTimeout(5000);
        conn.setDoOutput(true);
        conn.setDoInput(true);
        OutputStream outputStream = conn.getOutputStream();
        outputStream.write(postData.getBytes("UTF-8"));
        outputStream.flush();
        outputStream.close();
        String result = readStream(conn.getInputStream());
        conn.disconnect();
        return result;
    }
}

```

```
    }  
    catch (Exception e) {  
        Log.i("PHPRequest", "request was failed.");  
        return null;  
    }  
}  
}
```

12) 서버의 데이터베이스를 조작할 php 파일

■ php파일

addfriend.php

```
<?php
    header("Content-Type: text/html;charset=UTF-8");
    $conn = mysqli_connect("localhost","root","586394","appdata");
    $data_stream =
    "".$_POST['friendid'].",".$_POST['me_name'].",".$_POST['me_userid'].",".$_POST['friend_name'].",".$_POST['friend_userid'].",".$_POST['friend_image'].",";
    $query = "insert into
    friend_relation(friendid,me_name,me_userid,friend_name,friend_userid,friend_image)
    values ( ".$data_stream.")";
    $result = mysqli_query($conn, $query);

    if($result)
        echo "1";
    else
        echo "-1";

    mysqli_close($conn);
?>
```

addgroup.php

```
<?php
    header("Content-Type: text/html;charset=UTF-8");
    $conn = mysqli_connect("localhost","root","586394","appdata");
    $data_stream = "".$_POST['groupname'].",".$_POST['grouppassword'].",";
    $query = "insert into group_list(group_name,group_password) values
    ( ".$data_stream.")";
    $result = mysqli_query($conn, $query);

    if($result)
        echo "1";
    else
        echo "-1";

    mysqli_close($conn);
?>
```

addmarker.php

```
<?php
    header("Content-Type: text/html;charset=UTF-8");
    $conn = mysqli_connect("localhost","root","586394","appdata");
    $data_stream =
    "".$_POST['markerid'].",".$_POST['lat'].",".$_POST['lng'].",".$_POST['name_place'].",".$_POST['name_group'].",".$_POST['content'].",".$_POST['name_user'].",".$_POST
```



```

['userid'].''";
$query = "insert into
marker(markerid,lat,lng,name_place,name_group,content,name_user,userid) values
('.$data_stream.')";
$result = mysqli_query($conn, $query);

if($result)
    echo "1";
else
    echo "-1";

mysqli_close($conn);
?>

```

adduserintogroup.php

```

<?php
header("Content-Type: text/html;charset=UTF-8");
$conn = mysqli_connect("localhost","root","586394","appdata");
$data_stream =
"".$_POST['relationid'].','.$_POST['groupname'].','.$_POST['userid'].''";
$query = "insert into group_relation(relationid,group_name,userid) values
('.$data_stream.')";
$result = mysqli_query($conn, $query);

if($result)
    echo "1";
else
    echo "-1";

mysqli_close($conn);
?>

```

delete.php

```

<?php
header("Content-Type: text/html;charset=UTF-8");
$conn = mysqli_connect("localhost","root","586394","appdata");
$data_stream = "".$_POST['userid'].''";
$query = "delete from user where userid=$data_stream";
$result = mysqli_query($conn, $query);

if($result)
    echo "1";
else
    echo "-1";

mysqli_close($conn);
?>

```

deletefriend.php

```

<?php
    header("Content-Type: text/html;charset=UTF-8");
    $conn = mysqli_connect("localhost","root","586394","appdata");
    $data_stream = "".$_POST['friendid']. "";
    $query = "delete from friend_relation where friendid=$data_stream";
    $result = mysqli_query($conn, $query);

    if($result)
        echo "1";
    else
        echo "-1";

    mysqli_close($conn);
?>

```

deletegroup.php

```

<?php
    header("Content-Type: text/html;charset=UTF-8");
    $conn = mysqli_connect("localhost","root","586394","appdata");
    $data_stream = "".$_POST['relationid']. "";
    $query = "delete from group_relation where relationid=$data_stream";
    $result = mysqli_query($conn, $query);

    if($result)
        echo "1";
    else
        echo "-1";

    mysqli_close($conn);
?>

```

deletemarker.php

```

<?php
    header("Content-Type: text/html;charset=UTF-8");
    $conn = mysqli_connect("localhost","root","586394","appdata");
    $data_stream = "".$_POST['markerid']. "";
    $query = "delete from marker where markerid=$data_stream";
    $result = mysqli_query($conn, $query);

    if($result)
        echo "1";
    else
        echo "-1";

    mysqli_close($conn);
?>

```

getfriend.php

```

<?php

$link=mysqli_connect("localhost","root","586394", "appdata" );

```

```

if (!$link)
{
    echo "MySQL 접속 에러 : ";
    echo mysqli_connect_error();
    exit();
}

mysqli_set_charset($link,"utf8");
$me_userid=$_REQUEST[me_userid];

$sql="select friend_name,friend_userid,friend_image from friend_relation where
me_userid='$me_userid'";

$result=mysqli_query($link,$sql);
$data = array();
if($result){

    while($row=mysqli_fetch_array($result)){
        array_push($data,
            array('friend_name'=>$row[0],
'friend_userid'=>$row[1],
'friend_image'=>$row[2]
            ));
    }

    header('Content-Type: application/json; charset=utf8');
    $json = json_encode(array("kimhyju"=>$data),
JSON_PRETTY_PRINT+JSON_UNESCAPED_UNICODE);
    echo $json;

}
else{
    echo "SQL문 처리중 에러 발생 : ";
    echo mysqli_error($link);
}

mysqli_close($link);

?>

```

getfriendinfo.php

```

<?php

$link=mysqli_connect("localhost","root","586394", "appdata" );
if (!$link)

```

```

{
    echo "MySQL 접속 에러 : ";
    echo mysqli_connect_error();
    exit();
}

mysqli_set_charset($link,"utf8");
$userid=$_REQUEST[userid];

$sql="select imgurl,name from user where userid='$userid'";

$result=mysqli_query($link,$sql);
$data = array();
if($result){

    while($row=mysqli_fetch_array($result)){
        array_push($data,
            array('imgurl'=>$row[0],
                'name'=>$row[1]
            ));
    }

    header('Content-Type: application/json; charset=utf8');
    $json = json_encode(array("kimhyju"=>$data),
        JSON_PRETTY_PRINT+JSON_UNESCAPED_UNICODE);
    echo $json;

}
else{
    echo "SQL문 처리중 에러 발생 : ";
    echo mysqli_error($link);
}

mysqli_close($link);

?>

```

getgroup.php

```

<?php

$link=mysqli_connect("localhost","root","586394", "appdata" );
if (!$link)
{
    echo "MySQL 접속 에러 : ";
    echo mysqli_connect_error();
}

```

```

    exit();
}

mysqli_set_charset($link,"utf8");
$userid=$_REQUEST[userid];

$sql="select relationid,group_name from group_relation where userid='$userid'";

$result=mysqli_query($link,$sql);
$data = array();
if($result){

    while($row=mysqli_fetch_array($result)){
        array_push($data,
            array('relationid'=>$row[0],
'group_name'=>$row[1]
            ));
    }

    header('Content-Type: application/json; charset=utf8');
    $json = json_encode(array("kimhyju"=>$data),
JSON_PRETTY_PRINT+JSON_UNESCAPED_UNICODE);
    echo $json;

}
else{
    echo "SQL문 처리중 에러 발생 : ";
    echo mysqli_error($link);
}

mysqli_close($link);

?>

```

getgrouppassword.php

```

<?php

$link=mysqli_connect("localhost","root","586394", "appdata" );
if (!$link)
{
    echo "MySQL 접속 에러 : ";
    echo mysqli_connect_error();
    exit();
}

```

```

mysqli_set_charset($link,"utf8");
$group_name=$_REQUEST[group_name];

$sql="select group_password from group_list where group_name='$group_name'";

$result=mysqli_query($link,$sql);
$data = array();
if($result){

    while($row=mysqli_fetch_array($result)){
        array_push($data,
            array('group_password'=>$row[0]
        ));
    }

    header('Content-Type: application/json; charset=utf8');
    $json = json_encode(array("kimhyju"=>$data),
JSON_PRETTY_PRINT+JSON_UNESCAPED_UNICODE);
    echo $json;

}
else{
    echo "SQL문 처리중 에러 발생 : ";
    echo mysqli_error($link);
}

mysqli_close($link);

?>

```

getmarker.php

```

<?php

$link=mysqli_connect("localhost","root","586394", "appdata" );
if (!$link)
{
    echo "MySQL 접속 에러 : ";
    echo mysqli_connect_error();
    exit();
}

mysqli_set_charset($link,"utf8");
$name_group=$_REQUEST[name_group];

```

```
$sql="select markerid, lat, lng, name_place, content, name_user from marker where
name_group='$name_group'";
```

```
$result=mysqli_query($link,$sql);
```

```
$data = array();
```

```
if($result){
```

```
    while($row=mysqli_fetch_array($result)){
```

```
        array_push($data,
```

```
            array('markerid'=>$row[0],
```

```
'lat'=>$row[1],
```

```
'lng'=>$row[2],
```

```
'name_place'=>$row[3],
```

```
'content'=>$row[4],
```

```
'name_user'=>$row[5]
```

```
        ));
```

```
    }
```

```
    header('Content-Type: application/json; charset=utf8');
```

```
    $json = json_encode(array("kimhyju"=>$data),
```

```
    JSON_PRETTY_PRINT+JSON_UNESCAPED_UNICODE);
```

```
    echo $json;
```

```
}
```

```
else{
```

```
    echo "SQL문 처리중 에러 발생 : ";
```

```
    echo mysqli_error($link);
```

```
}
```

```
mysqli_close($link);
```

```
?>
```

getmarkerinfo.php

```
<?php
```

```
$link=mysqli_connect("localhost","root","586394", "appdata" );
```

```
if (!$link)
```

```
{
```

```
    echo "MySQL 접속 에러 : ";
```

```
    echo mysqli_connect_error();
```

```
    exit();
```

```
}
```

```
mysqli_set_charset($link,"utf8");
```

```

$markerid=$_REQUEST[markerid];

$sql="select name_place, content, name_user from marker where markerid='$markerid'";

$result=mysqli_query($link,$sql);
$data = array();
if($result){

    while($row=mysqli_fetch_array($result)){
        array_push($data,
            array('name_place'=>$row[0],
'content'=>$row[1],
'name_user'=>$row[2]

        ));
    }

    header('Content-Type: application/json; charset=utf8');
    $json = json_encode(array("kimhyju"=>$data),
JSON_PRETTY_PRINT+JSON_UNESCAPED_UNICODE);
    echo $json;

}
else{
    echo "SQL문 처리중 에러 발생 : ";
    echo mysqli_error($link);
}

mysqli_close($link);

?>

```

insert.php

```

<?php
    header("Content-Type: text/html;charset=UTF-8");
    $conn = mysqli_connect("localhost","root","586394","appdata");
    $data_stream = "".$_POST['userid'].",".$_POST['imgurl'].",".$_POST['name']."";
    $query = "insert into user(userid,imgurl,name) values (". $data_stream.")";
    $result = mysqli_query($conn, $query);

    if($result)
        echo "1";
    else
        echo "-1";

    mysqli_close($conn);

?>

```


2. 나의 앱 만들기 후기

■ 강민기



■나의 소감

이번 학기에 ‘모바일 프로젝트’ 강의를 들으면서 사실은 이해하지 못한 부분도 많이 있고 아직도 안드로이드 스튜디오를 이용해 앱을 개발한다는 것은 어렵게 느껴진다. 하지만 이번 기말 프로젝트를 준비하면서 한 계단 위로 올라갔다고 생각한다. 수업 중 배운 내용과 기말 프로젝트를 준비하면서 사용한 기능 외에도 훨씬 더 많은 기능들이 있는데 이런 것들도 다른 앱을 개발해보면서 다양하게 경험해보고 싶다.

■모바일 프로그래밍 강의를 통해 얻은 것

이번 강의를 통해서 확실하게 얻은 것은 내가 많이 부족하다는 점을 알았다는 것이다. 구체적이지는 않지만 개발자를 목표로 하고 있었는데 어플 개발자가 되기에는 정말 턱 없이 부족한 실력이다. 하지만 부족함을 알았으니 앞으로 이 수업이 끝나도 모바일 프로그래밍을 더 공부해봐야겠다. 방학때도 어플리케이션 개발에 시간을 투자해서 다양한 앱을 만들어 보고싶다. 사용자들에게 사용하기 쉬우면서도 유용하고 UI/UX도 밀리지 않는 그런 앱을 생각해서 개발을 시도해 볼 생각이다. 또 기회가 된다면 수업시간에 배웠던 IoT와 라즈베리파이도 공부해서 직접 해보고 싶다. 끝으로 한 학기동안 강의해주신 교수님도 수고하셨습니다.

■ 김형준



■나의 소감

1, 2학년 때 배운 프로그래밍들은 대부분 UI의 구성없이 DOS창 환경에서 실행되었다. 그러나 이번 모바일 프로그래밍 수업을 배우면서 내가 직접 UI를 배치하고 그에따라 java파일을 코딩하였다. 아주 기본적이고 간단한 어플리케이션이지만 바로바로 스마트폰에서 결과물이 실행되니깐 성취감이 다른 프로그래밍 수업보다 월등히 뛰어났다. 그러므로 다른 프로그래밍 수업보다 학습하거나 배우는 면에서 훨씬 흥미가 있었다. 이번 기말 프로젝트를 통해서 자바에 대해 더 깊게 공부할 수 있었고 어떻게 하면 더 효율적으로 앱을 구성할 수 있을까 많이 고민했었다. 한번 알아서 코딩을 시작하다보면 시간가는줄 모르고 코딩하다가 항상 세네시에 잠들었다. 또한 서버와 데이터베이스를 구현하기 위해 많은 정보를 검색해 보았고 실제로 구현해보면서 서버에 대한 기본적인 원리를 이해하였다. 기말프로젝트를 통해서 모바일 프로그래밍 외에도 많은 것을 배울 수 있어서 굉장히 뜻깊은 시간이었다.

■모바일 프로그래밍 강의를 통해 얻은 것

비록 모바일 프로그래밍에서 기초적인 수준의 내용을 배웠지만 실습을 통해 스마트폰에서 어플리케이션이 실행되는 것을 보고 매 수업마다 보람을 느낄 수 있는 수업이었고 C, C++, JAVA 강의와 달리 직접 구성한 UI를 통해 프로그램이 실행되는 것을 보면서 흥미를 가지고 배울 수 있었다. 안드로이드는 자바로 실행되는 환경이라서 모바일 프로그래밍 강의를 들으면서 자바에 대해 많은 부분을 알 수 있었고 특히 이번 기말 프로젝트를 진행하면서 카카오톡으로 로그인, 구글지도, 구글 장소 자동완성, 서버, 데이터베이스를 구현하기 위해 많은 검색을 하고 배웠다. 대부분의 소스들은 몇번의 구글링을 통해 구할 수 있어지만 내가 만들고 있는 앱과 연결시키는 과정이 조금 까다로웠다. 그러나 여러가지 방면으로 시도를 해보다가 결국 성공적으로 실행되면 보람과 성취감을 느꼈으며 조금이라도 더 효율적으로 구현하기 위해 고민하는 것이 즐거웠다. 모바일 프로그래밍 강의를 통해서 얻은 것 중 강의시간에 얻는 것도 많지만 기말프로젝트를 하면서 직접 검색을 통해 얻은 것이 더 많은 것 같다. 원하는 정보를 얻기위해 검색하는 법, 서버를 통해 데이터베이스를 조작하는 법, 서버에서 얻은 정보를 JSON을 이용해 데이터로 추출하는 법등 다양한 프로그래밍 기술을 배웠다. 이번 학기 모바일 프로그래밍 강의를 듣고 기말프로젝트를 진행하면서 나 자신을 한단계 발전시키는 계기가 될 수 있었다.