안드로이드 기반
동아리 애플리케이션 제작

1. 작품 요약

- 현저히 떨어진 인터넷 홈페이지 접근성을 높이기 위해 애플리케이션 제작 결정
- 안드로이드 기반으로 제작
- 마켓에 업로드하여 누구든지 다운받아 사용할 수 있게 제작

2. 기능

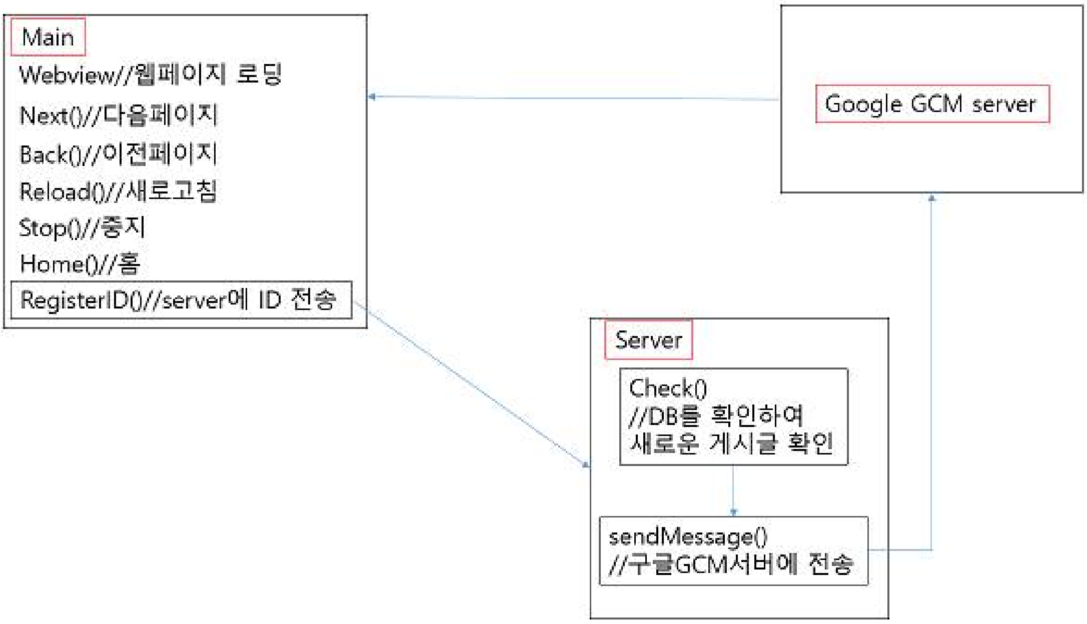
- 홈페이지 접속
- 뒤로가기, 앞으로가기, 새로고침 등 기존 인터넷 브라우저에서 사용한 도구 탑재
- 공지 메신저에 공지하기 위해 해당 페이지 URL복사 기능(추후 기능 업데이트 예정)
- 빠른 회원가입 메뉴 접속

3. 기대 효과

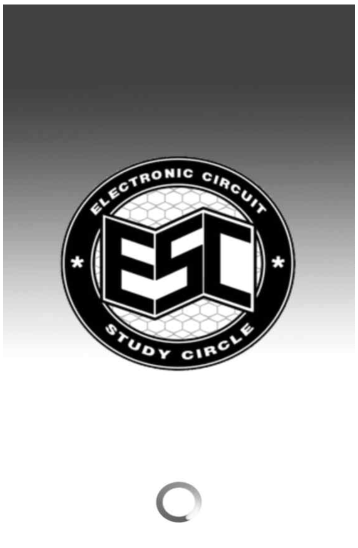
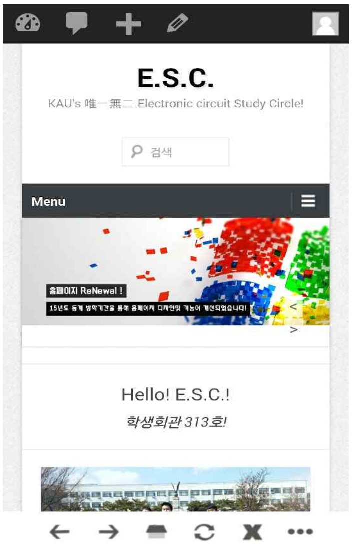
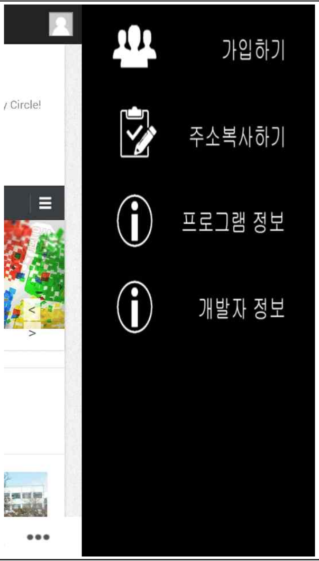
- 인터넷 홈페이지 보다 좀 더 향상된 접근성으로 방문이 잦아지고 원활한 선, 후배간의 소통이 이루어 질 것을 기대

4. 세부사항

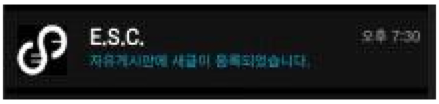
① 전체적인 구조



② 디자인

(1) 초기	(2) 메인	(3) 메뉴
 	 E.S.C. KAU's 唯一無二 Electronic circuit Study Circle! 검색 Menu 홈페이지 ReNewal ! 15년도 동계 방학기간을 통해 홈페이지 디자인및 기능이 개선되었습니다! Hello! E.S.C.! *학생회관 313호!*	

③ 푸시알림



E.S.C. 오후 7:30

자유게시판에 새글이 등록되었습니다.

④ 소스코드

* Client

○ MainActivity.java

```
package project.esc;

public class MainActivity extends Activity {

    WebView wv;
    private ValueCallback<Uri> mUploadMessage;
    private final static int FILECHOOSER_RESULTCODE = 1;
    public static final String EXTRA_MESSAGE = "message";
    public static final String PROPERTY_REG_ID = "registration_id";
    private static final String PROPERTY_APP_VERSION = "appVersion";
    private static final int PLAY_SERVICES_RESOLUTION_REQUEST = 9000;

    private BackPressCloseHandler handler;
    String SENDER_ID = "000000000000";
    GoogleCloudMessaging gcm;
    Context context;
    String regid;
    static final String TAG = "GCM";

    //////////////////////////////////////////////////////////////////////////////////////////////////////////////

    private DisplayMetrics metrics;
    private LinearLayout slidingPanel;
    private RelativeLayout rightMenuPanel;
    private FrameLayout.LayoutParams slidingPanelParameters;
    private FrameLayout.LayoutParams rightMenuPanelParameters;
    private int panelWidth;
    public static boolean isRightExpanded;

    protected void onActivityResult(int requestCode, int resultCode,
                Intent intent) {
        if (requestCode == FILECHOOSER_RESULTCODE) {
                if (mUploadMessage == null) {
                        return;
                }
```

```
            Uri result = (intent == null || resultCode != RESULT_OK) ? null
                                    : intent.getData();

            mUploadMessage.onReceiveValue(result);

            mUploadMessage = null;
        }
}

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    startActivity(new Intent(this, SplashActivity.class));//Splash화면

    //웹뷰 설정
    wv = (WebView) findViewById(R.id.webview);
    wv.loadUrl("http://esc.kau.ac.kr");
    wv.setWebViewClient(new SiteWebViewClient());
    wv.getSettings().setJavaScriptEnabled(true);
    wv.getSettings().setPluginState(PluginState.ON);
    wv.getSettings().setAppCacheEnabled(true);
    wv.getSettings().setCacheMode(wv.getSettings().LOAD_NO_CACHE);

    //뒤로가기 버튼
    Button back = (Button) findViewById(R.id.back);
    back.setOnClickListener(new OnClickListener() {
            public void onClick(View v) {
                    wv.goBack();
            }
    });

    //앞으로가기 버튼
    Button next = (Button) findViewById(R.id.next);
    next.setOnClickListener(new OnClickListener() {
            public void onClick(View v) {
                    wv.goForward();
            }
    });
```

```
//홈 버튼
Button home = (Button) findViewById(R.id.home);
home.setOnClickListener(new OnClickListener() {
        public void onClick(View v) {

                wv.loadUrl("http://esc.kau.ac.kr");
        }
});

//새로고침 버튼
Button reload = (Button) findViewById(R.id.reload);
reload.setOnClickListener(new OnClickListener() {
        public void onClick(View v) {
                wv.reload();
        }
});

//중지 버튼
Button stop = (Button) findViewById(R.id.stop);
stop.setOnClickListener(new OnClickListener() {
        public void onClick(View v) {
                wv.stopLoading();
        }
});

//기타(설정) 버튼
Button etc = (Button) findViewById(R.id.etc);
etc.setOnClickListener(new OnClickListener() {
        public void onClick(View v) {
                menuRightSlideAnimationToggle();

        }
});

handler = new BackPressCloseHandler(this);//백키 종료 핸들러

context = getApplicationContext();

//구글플레이 서비스 확인
```

```
if (checkPlayServices()) {
    gcm = GoogleCloudMessaging.getInstance(this);
    regid = getRegistrationId(context);
    Log.e("RegID", regid);
    if (regid.isEmpty()) {

        registerInBackground();
    }
} else {

}



//기타(설정) 버튼 눌렀을 시 효과 초기화

metrics = new DisplayMetrics();
getWindowManager().getDefaultDisplay().getMetrics(metrics);
panelWidth = (int) ((metrics.widthPixels) * 0.75);
slidingPanel = (LinearLayout) findViewById(R.id.sliding);
slidingPanelParameters = (FrameLayout.LayoutParams) slidingPanel
        .getLayoutParams();
slidingPanelParameters.width = metrics.widthPixels;
slidingPanel.setLayoutParams(slidingPanelParameters);

// right slide menu initialize
rightMenuPanel = (RelativeLayout) findViewById(R.id.rightMenuPanel);
rightMenuPanelParameters = (FrameLayout.LayoutParams) rightMenuPanel
        .getLayoutParams();
rightMenuPanelParameters.width = panelWidth;
rightMenuPanel.setLayoutParams(rightMenuPanelParameters);

Button register = (Button) findViewById(R.id.register);
Button share = (Button) findViewById(R.id.share);
Button program_info = (Button) findViewById(R.id.program_info);
Button developer_info = (Button) findViewById(R.id.developer_info);

//기타(설정) 슬라이드 내 회원가입 버튼
register.setOnClickListener(new OnClickListener() {

    @Override
```

```
            public void onClick(View v) {
                    // TODO Auto-generated method stub
                    isRightExpanded = false;

                    // Collapse
                    new CloseAnimation(slidingPanel, panelWidth,
                                    TranslateAnimation.RELATIVE_TO_SELF, -0.73f,
                                    TranslateAnimation.RELATIVE_TO_SELF, 0.0f,
0, 0.0f, 0, 0.0f);

                    // enable all of main view
                    FrameLayout viewGroup = (FrameLayout)
findViewById(R.id.view)
                                    .getParent();
                    enableDisableViewGroup(viewGroup, true);

                    // disable empty view
                    ((LinearLayout)
findViewById(R.id.empty)).setVisibility(View.GONE);
                    findViewById(R.id.empty).setEnabled(false);
                    wv.loadUrl("http://esc.kau.ac.kr/wp-login.php?action=register");
            }

    });

    //기타(설정) 슬라이드 내 공유하기 버튼
    share.setOnClickListener(new OnClickListener() {

            @Override
            public void onClick(View v) {
                    // TODO Auto-generated method stub
                    ClipboardManager clipboard = (ClipboardManager)
getSystemService(Context.CLIPBOARD_SERVICE);
                    ClipData clip = ClipData.newPlainText("label",wv.getUrl());
                    clipboard.setPrimaryClip(clip);
                    Toast.makeText(getApplicationContext(), "주소가 클립보드에
복사되었습니다!", Toast.LENGTH_SHORT).show();



            }
    });

    //기타(설정) 슬라이드 내 프로그램정보 버튼
    program_info.setOnClickListener(new OnClickListener() {

            @Override
            public void onClick(View v) {
                    // TODO Auto-generated method stub
                    AlertDialog p_info = create_inputDialog(Html
                                    .fromHtml("E.S.C. Homepage" +
"<br>version : 2.0v"
                                                    + "<br>COPYRIGHT 2015
E.S.C."
                                                    + "<br>All rights
reserved."));
                    p_info.show();
            }

    });

    //기타(설정) 슬라이드 내 개발자정보 버튼
    developer_info.setOnClickListener(new OnClickListener() {

            @Override
            public void onClick(View v) {
                    // TODO Auto-generated method stub
                    AlertDialog d_info =
create_inputDialog(Html.fromHtml("E.S.C."
                                    + "<br>Application: 32th
Park.M.J."+"<br>Homepage Design: 36th Jang.J.Y&Kim.Y.J."
                                    + "<br>contact :
zzangakswns@gmail.com"));
                    d_info.show();
            }

    });


}
```

```
//백키 두번 눌렀을 시 종료
public boolean onKeyDown(int KeyCode, KeyEvent event) {
    if ((KeyCode == KeyEvent.KEYCODE_BACK) &&
wv.getUrl().equals("http://esc.kau.ac.kr/")) {
            handler.onBackPressed();
            return true;
    } else if ((KeyCode == KeyEvent.KEYCODE_BACK)
                    && wv.canGoBack()) {
            wv.goBack();

            return true;
    }
    return super.onKeyDown(KeyCode, event);
}

private class SiteWebViewClient extends WebViewClient {
    public boolean shouldOverrideUrlLoading(WebView view, String url) {
            view.loadUrl(url);
            return true;
    }
}

//프로그램정보, 개발자정보 다이얼로그 플랫폼
private AlertDialog create_inputDialog(CharSequence message) {
    AlertDialog dialogBox = new AlertDialog.Builder(this)
                    .setTitle("E.S.C.")
                    .setIcon(R.drawable.ic_launcher)
                    .setMessage(message)
                    .setPositiveButton("Check",
                                    new DialogInterface.OnClickListener() {
                                            public void onClick(DialogInterface
dialog,
                                                            int which) {
                                                    dialog.cancel();
                                            }
                                    })
                    .setNegativeButton("Cancel",
                                    new DialogInterface.OnClickListener() {
```

```
                                              public void onClick(DialogInterface
dialog,
                                                              int which) {
                                                      dialog.cancel();

                                              }
                                     }).create();

    return dialogBox;
}


@Override
protected void onResume() {

    super.onResume();

    checkPlayServices();

}


//구글 서비스 체크 함수
private boolean checkPlayServices() {

    int resultCode = GooglePlayServicesUtil
                    .isGooglePlayServicesAvailable(this);

    if (resultCode != ConnectionResult.SUCCESS) {

            if (GooglePlayServicesUtil.isUserRecoverableError(resultCode)) {

                    GooglePlayServicesUtil.getErrorDialog(resultCode, this,

                    PLAY_SERVICES_RESOLUTION_REQUEST).show();

            } else {

                    Log.i(TAG, "This device is not supported.");
```

```
                        finish();

                }

                return false;

        }

        return true;

}

//GCM에 보낼 RegistrationId 얻음
private String getRegistrationId(Context context) {

        final SharedPreferences prefs = getGcmPreferences(context);

        String registrationId = prefs.getString(PROPERTY_REG_ID, "");

        if (registrationId.isEmpty()) {

                Log.i(TAG, "Registration not found.");

                return "";

        }

        int registeredVersion = prefs.getInt(PROPERTY_APP_VERSION,
                        Integer.MIN_VALUE);

        int currentVersion = getAppVersion(context);

        if (registeredVersion != currentVersion) {

                Log.i(TAG, "App version changed.");

                return "";

        }
```

```
    return registrationId;

}

private static int getAppVersion(Context context) {

    try {

            PackageInfo packageInfo = context.getPackageManager()

            .getPackageInfo(context.getPackageName(), 0);

            return packageInfo.versionCode;

    } catch (NameNotFoundException e) {

            throw new RuntimeException("Could not get package name: " + e);

    }

}


private SharedPreferences getGcmPreferences(Context context) {

    return getSharedPreferences(MainActivity.class.getSimpleName(),

    Context.MODE_PRIVATE);

}

//3rd Party와 연동 실행
private void registerInBackground() {
    new RegistrationThread().start();
}

//GCM에 보낼 RegistrationId를 3rd Party 서버에 보냄
private void sendRegistrationIdToBackend() {
    try {

            String addr = "http://esc.kau.ac.kr/______.php?u_id=" + regid
                              + "";
            URL url = new URL(addr);

            url.openStream();
            Log.e("u_id", addr);
    } catch (MalformedURLException e) {

            // TODO Auto-generated catch block

            e.printStackTrace();

    } catch (IOException e) {

            // TODO Auto-generated catch block

            e.printStackTrace();

    }
}

@Override
protected void onDestroy() {

    super.onDestroy();

}

private void storeRegistrationId(Context context, String regId) {

    final SharedPreferences prefs = getGcmPreferences(context);

    int appVersion = getAppVersion(context);

    Log.i(TAG, "Saving regId on app version " + appVersion);

    SharedPreferences.Editor editor = prefs.edit();
    editor.putString(PROPERTY_REG_ID, regId);

    editor.putInt(PROPERTY_APP_VERSION, appVersion);

    editor.commit();

}

//3rd Party 서버 연동
class RegistrationThread extends Thread {
    public void run() {
        try {
            if (gcm == null) {
                gcm = GoogleCloudMessaging.getInstance(context);
            }
            regid = gcm.register(SENDER_ID);

            sendRegistrationIdToBackend();
        } catch (Exception e) {
            e.getStackTrace();

        }
    }
}

//기타(설정)창 애니메이션
public void menuRightSlideAnimationToggle() {
    if (!isRightExpanded) {
        isRightExpanded = true;
        rightMenuPanel.setVisibility(View.VISIBLE);

        new ExpandAnimation(slidingPanel, panelWidth, "right",
                Animation.RELATIVE_TO_SELF, 0.0f,
                Animation.RELATIVE_TO_SELF, -0.73f, 0, 0.0f, 0, 0.0f);

        FrameLayout viewGroup = (FrameLayout) findViewById(R.id.view)
                .getParent();
        enableDisableViewGroup(viewGroup, false);

        ((LinearLayout) findViewById(R.id.empty))
```

```
                            .setVisibility(View.VISIBLE);

            findViewById(R.id.empty).setEnabled(true);
            findViewById(R.id.empty).setOnTouchListener(new OnTouchListener() {


                    @Override
                    public boolean onTouch(View v, MotionEvent event) {
                            // TODO Auto-generated method stub
                            menuRightSlideAnimationToggle();
                            return true;
                    }
            });

    } else {
            isRightExpanded = false;

            new CloseAnimation(slidingPanel, panelWidth,
                            TranslateAnimation.RELATIVE_TO_SELF, -0.75f,
                            TranslateAnimation.RELATIVE_TO_SELF, 0.0f, 0, 0.0f, 0,
0.0f);

            FrameLayout viewGroup = (FrameLayout) findViewById(R.id.view)
                            .getParent();
            enableDisableViewGroup(viewGroup, true);

            ((LinearLayout) findViewById(R.id.empty)).setVisibility(View.GONE);
            findViewById(R.id.empty).setEnabled(false);
    }
}

public static void enableDisableViewGroup(ViewGroup viewGroup,
            boolean enabled) {
    int childCount = viewGroup.getChildCount();
    for (int i = 0; i < childCount; i++) {
            View view = viewGroup.getChildAt(i);

            if (view.getId() != R.id.etc) {
                    view.setEnabled(enabled);
                    if (view instanceof ViewGroup) {
```

```
                                enableDisableViewGroup((ViewGroup) view, enabled);
                        }
                }
        }
    }


}
```

○ **GCMIntentService.java**

```
package project.esc;

public class GcmIntentService extends IntentService {
    public static final int NOTIFICATION_ID = 2;
    private NotificationManager mNotificationManager;
    NotificationCompat.Builder builder;

    public GcmIntentService() {
        super("GcmIntentService");
    }
    public static final String TAG = "GCM";

    MainActivity main = new MainActivity();


    @Override
    protected void onHandleIntent(Intent intent) {
        Bundle extras = intent.getExtras();
        GoogleCloudMessaging gcm = GoogleCloudMessaging.getInstance(this);

        String messageType = gcm.getMessageType(intent);

        if (!extras.isEmpty()) {

            if
(GoogleCloudMessaging.MESSAGE_TYPE_SEND_ERROR.equals(messageType)) {
                sendNotification("Send error: " + extras.toString());
            }                    else                    if
(GoogleCloudMessaging.MESSAGE_TYPE_DELETED.equals(messageType)) {
```

```
                    sendNotification("Deleted messages on server: " +
extras.toString());

               } else if
(GoogleCloudMessaging.MESSAGE_TYPE_MESSAGE.equals(messageType)) {

               sendNotification(extras.getString("msg"));
               Log.e(TAG, "Received: " + extras.toString());

               }
          }
          // Release the wake lock provided by the WakefulBroadcastReceiver.
          GCMBroadcastReceiver.completeWakefulIntent(intent);
     }

   //받은 푸시 알림 출력 메소드
     private void sendNotification(String msg) {
          mNotificationManager = (NotificationManager)
                    this.getSystemService(Context.NOTIFICATION_SERVICE);



          PendingIntent contentIntent = PendingIntent.getActivity(this, 0,
                    new Intent(this, MainActivity.class), 0);

          NotificationCompat.Builder mBuilder =
                    new NotificationCompat.Builder(this)
          .setSmallIcon(R.drawable.ic_launcher)
          .setContentTitle("E.S.C.")
          .setAutoCancel(true)
          .setStyle(new NotificationCompat.BigTextStyle()
          .bigText(msg))
          .setContentText(msg);

          mBuilder.setContentIntent(contentIntent);
          mNotificationManager.notify(NOTIFICATION_ID, mBuilder.build());
     }
}
```

○ **BackPressCloseHandler.java**

```
package project.esc;

public class BackPressCloseHandler {

	private long Time =0;
	private Toast toast;
	private Activity activity;

	public BackPressCloseHandler(Activity context){
		this.activity=context;
	}

	public void onBackPressed(){
		if(System.currentTimeMillis()>Time+2000){//2초 이내
			Time = System.currentTimeMillis();
			show();
			return;
		}
		else{
			activity.finish();
			toast.cancel();
		}
	}

	public void show(){
		toast = Toast.makeText(activity, "\'뒤로\'버튼 한번 더 누르시면 종료됩니다.", Toast.LENGTH_SHORT);
		toast.show();
	}

}
```

* 3rd Party Server

○ **connmysql.java**

```
package project.esc;

public class connmysql {
    private Statement stat;
    private ResultSet rs;
    private ArrayList<String> aList;

    //board_id와 그 제목을 가져오는 메소드
    public int separate(){
        String URL = "jdbc:mysql://localhost:3306/000?autoReconnect=true";
        String ID = "root";
        String password="0000000";
        ResultSet result;
        ArrayList<Integer> list = new ArrayList<Integer>();
        ArrayList<String> t_list = new ArrayList<String>();
        int board_id=0;
        String temp="";
        try{
                Connection connect = null;
                connect = DriverManager.getConnection(URL,ID,password);

                Statement state = connect.createStatement();



                result = state.executeQuery("SELECT board_id,date FROM
wp_kboard_board_content");
                while(result.next()){
                        list.add(result.getInt(1));
                        t_list.add(result.getString(2));

                }
                board_id = list.get(list.size()-1);
                System.out.println("list.size"+list.size());
                for(int i=0;i<list.size();i++){
                        System.out.println("list : "+list.get(i));
                System.out.println("t_list : "+t_list.get(i));
            }

            System.out.println("DB board_id : "+board_id);

            state.close();
            connect.close();

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

        return board_id;

    }

    //새글 확인 할 제목 불러오는 메소드
    public final String check(){
        String URL = "jdbc:mysql://localhost:3306/000?autoReconnect=true";
        String ID = "root";
        String password="000000";
        ResultSet result;
        ArrayList<String> list = new ArrayList<String>();
        String back="";

        try{
            Connection connect = null;
            connect = DriverManager.getConnection(URL,ID,password);

            Statement state = connect.createStatement();

            result = state.executeQuery("SELECT date FROM
wp_kboard_board_content");


            while(result.next()){
                list.add(result.getNString(1));
            }
            back = list.get(list.size()-1);
            state.close();
            connect.close();

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


        return back;

    }

    //push에 저장되어있는 휴대폰 ID 불러오는 메소드
    public final ArrayList<String> pushMySql() {
        String dbUrl = "jdbc:mysql://localhost:3306/push?autoReconnect=true";
        String id = "root";
        String pwd = "esc313!";

        try {
            aList = new ArrayList<String>();
            Connection con = null;
            con = DriverManager.getConnection(dbUrl, id, pwd);

            System.out.println("Success");
            stat = con.createStatement();


            // ResultSet
            rs = stat.executeQuery("SELECT * FROM push");

            // ResultSet
            while (rs.next()) {
                 aList.add(rs.getNString(1));
                 System.out.println(aList);
            }
            stat.close();
            con.close();

        } catch (SQLException sqex) {
            System.out.println("SQLException: " + sqex.getMessage());
```

```
            System.out.println("SQLState: " + sqex.getSQLState());
        }



        return aList;

    }
}
```

○ GCMServerSide.java

```
package project.esc;

import java.io.IOException;
import java.io.UnsupportedEncodingException;
import java.net.ConnectException;
import java.net.ServerSocket;
import java.net.URLEncoder;
import java.util.ArrayList;
import java.util.List;

import javax.naming.CommunicationException;

import com.google.android.gcm.server.Message;
import com.google.android.gcm.server.MulticastResult;
import com.google.android.gcm.server.Result;
import com.google.android.gcm.server.Sender;
import com.mysql.jdbc.CommunicationsException;

public class GCMServerSide implements Runnable {



        String encoder;
        connmysql a = new connmysql();
        ArrayList<String> list = a.pushMySql();
        String check = "1";//확인하는 게시판  제목
        String check_date=a.check(),new_check_date;//최신 게시글 날짜
        int board_id=0;
        String check_tmp="";
        ServerSocket serverSocket;
```

```
    public GCMServerSide(){
        try{
            serverSocket = new ServerSocket(3306);
            System.out.println("서버가 준비 되었습니다.");
        }catch(Exception e){
            e.getStackTrace();
        }
    }

    //푸시 알림으로 쓸 메세지 전송
    public void sendMessage(ArrayList<String> aList, int board_id) throws
IOException {

        Sender sender = new
Sender("0000000000000000000000000000000000000");
        List<String> list = aList;

        try{
            String message="";
            this.board_id=board_id;
            System.out.println("initial borad_id : "+board_id);
            if(board_id==1){
                System.out.println("borad_id : "+board_id);
                message = "자유게시판에 새글이 등록되었습니다.";
            }

            else if(board_id==3){
                System.out.println("borad_id : "+board_id);
                message = "OB게시판에 새글이 등록되었습니다.";
            }
            else if(board_id==11||board_id==5){
                System.out.println("borad_id : "+board_id);
                message = "공지사항에 새글이 등록되었습니다.";
            }
            else if(board_id==12){
                System.out.println("borad_id : "+board_id);
                message = "갤러리에 새글이 등록되었습니다.";
            }

            else if(board_id==10){
                System.out.println("borad_id : "+board_id);
                message = "회의록에 새글이 등록되었습니다.";
            }
            else if(board_id==9){
                System.out.println("borad_id : "+board_id);
                message = "FAQ에 새글이 등록되었습니다.";
            }
            else if(board_id==8){
                System.out.println("borad_id : "+board_id);
                message = "운영진소개에 새글이 등록되었습니다.";
            }
            else if(board_id==2){
                System.out.println("borad_id : "+board_id);
                message = "자료실에 새글이 등록되었습니다.";
            }
            else if(board_id==4){
                System.out.println("borad_id : "+board_id);
                message = "운영진게시판에 새글이 등록되었습니다.";
            }


            encoder = message;
        }catch(Exception e){
            e.getStackTrace();
        }


        System.out.println(encoder + "");
        Message messages = new Message.Builder().addData("msg",
encoder).build();
        MulticastResult multiResult = sender.send(messages, list, 5);


        if (multiResult != null) {

            List<Result> resultList = multiResult.getResults();
            System.out.println(resultList.size());

            for (Result result : resultList) {
                System.out.println(result.getMessageId());

            }

        }
        check_date=new_check_date;

    }

    public static void main(String[] args) throws Exception {

        GCMServerSide s = new GCMServerSide();

        s.run();

    }

    @Override
    public void run() {
        // TODO Auto-generated method stub
        while (true) {
            //새글 등록 확인 알고리즘
            try {
                new_check_date=a.check();
                if(new_check_date.equals(check_date)==false){
                    board_id = a.separate();
                    sendMessage(list,board_id);

System.out.println("check_Data"+check_date);
                }

            } catch (Exception ce) {
                ce.getMessage();
            }
        }
    }
}
```

※ 현재 운영 중인 프로그램이므로 보안상 문제가 될 수 있는 정보 '0'으로 대체