**Mobile Term Project**

**Part-Timer Scheduler**

**정유석, 박정호, 곽성훈**

**INDEX**

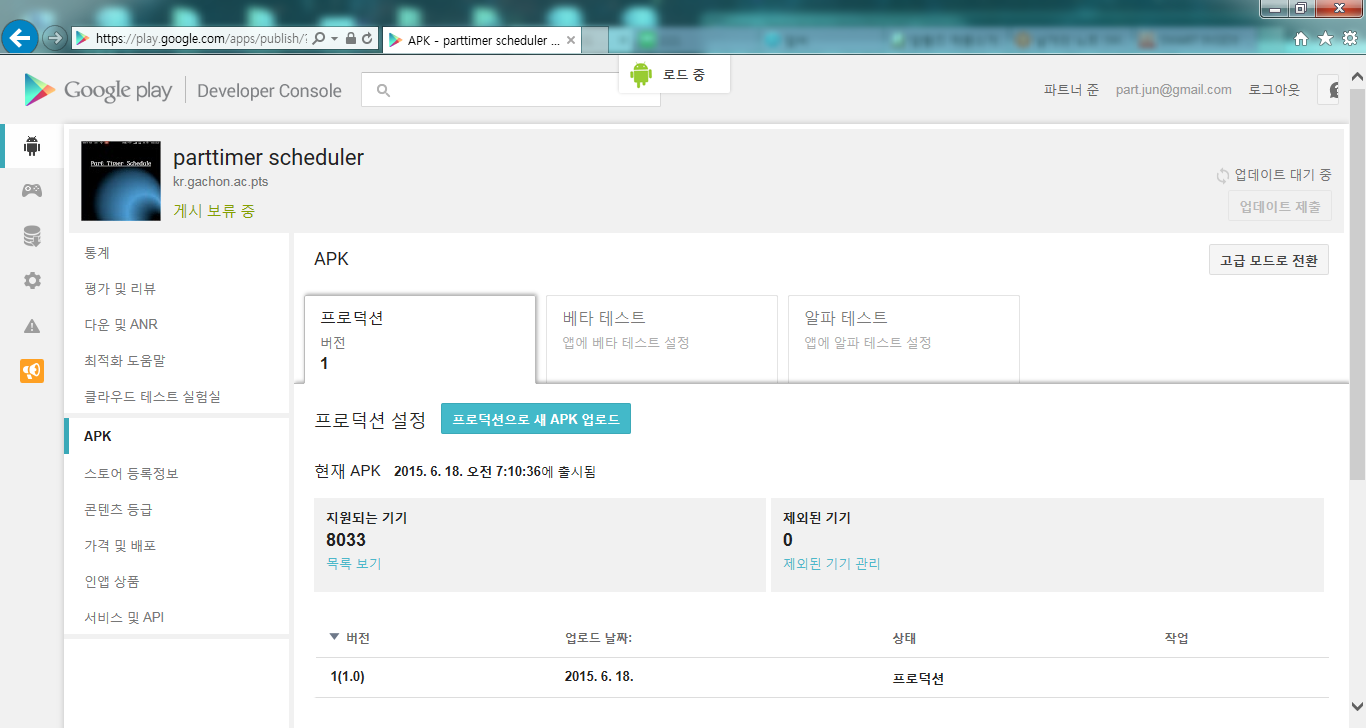
**1. In Google Play Store**

**2. Screenshot**

**3. System Code**

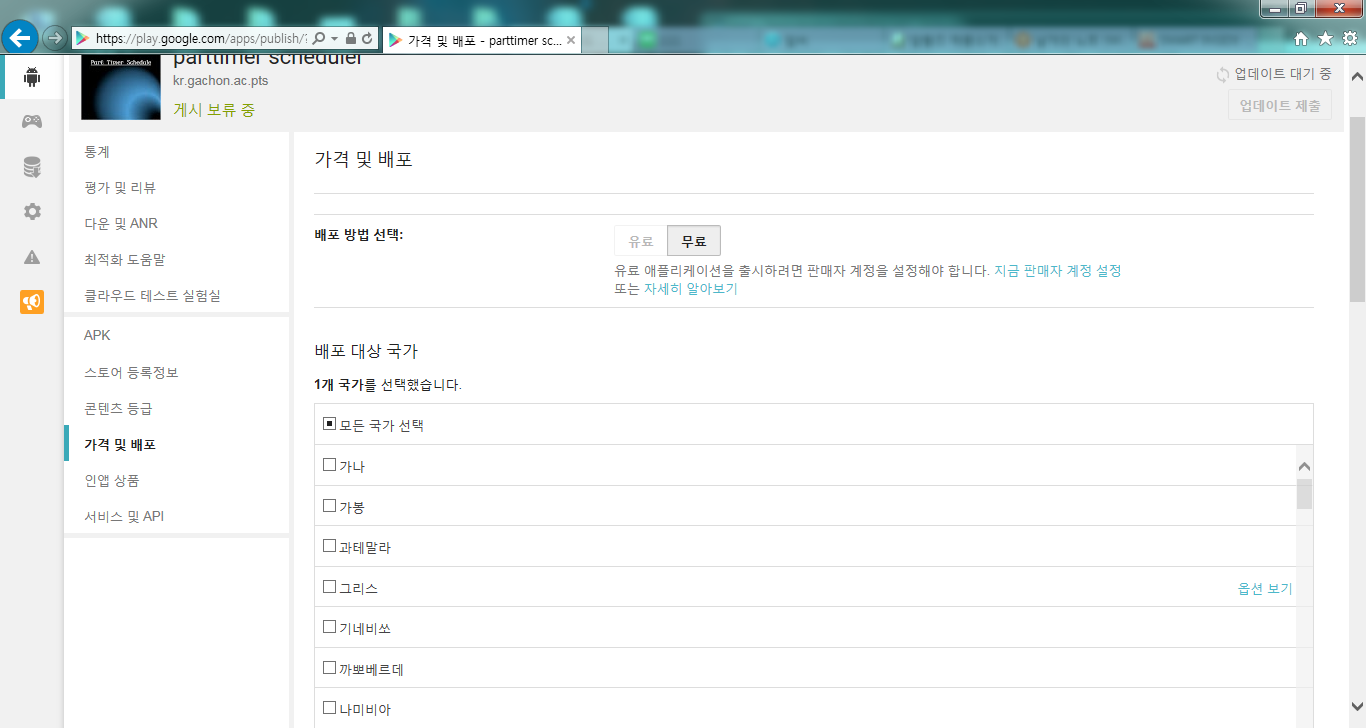
**1. In Google Play Store**

**- Process**

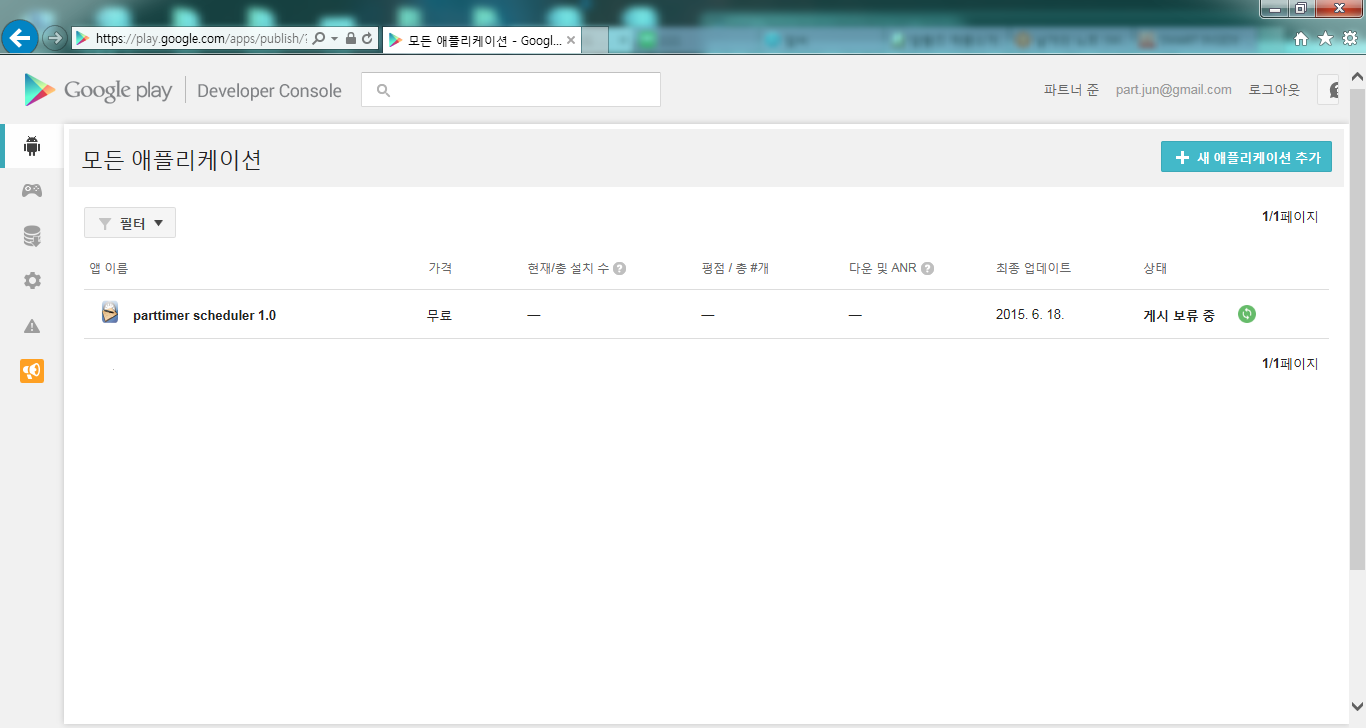








**- Current State**



**2. Screenshot**

**- Login Activity**



(1). User can login if user write correct ID & Password in DB

(2). If user write incorrect ID & Password in DB, System show error message

**- User Enrollment Activity**



(1). User can check ID duplication through duplication check button

(2). If user check manager check box, user can make new location Code.

Then, If location code already exist, user can’t make new location Code

(3). If user check part timer check box, user can select existent location Code

**- Menu Activity**



(1). If user login through manager mode, user can enter all of menu.

(2). If user login through part timer mode, before getting permission from manager, user can’t enter all of menu.

(3). After getting permission from manager, user can enter all of button except manager right button.

**-Apply Schedule Activity**



(1). User can register their preferred schedule if manager is accepting part timer

(2). If user register their schedule, it is stored in DB and their manager can check this applying.

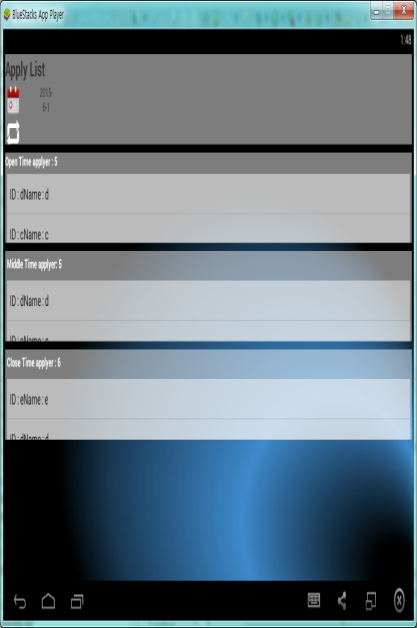
(3). User can check available data in upper box

**- bulletin Activity**



(1). User can use bulletin board. It is made by using Ex engine.

**- Fix Schedule Activity**



(1). manager can check part timer’s applying list.

(2). If manager click rotation button, rotation activity is active.

**- rotation Activity**



(1). Manager rotate applicant randomly

(2). If manager click rotation button, selected applicant appear in applicant list.

(3). And then, If user click commit button, fixed list is stored in DB and each part timer can check fixed schedule.

**- schedule enrollment Activity**



(1). Manager can register working data.

(2). Manager can select Start date and End date and number of worker in each days.

**- Accept new part timer Activity.**

<After click cancel button> <before>



(1).If new part timer register their information, manager can check people who apply enrollment in their location.

(2). If manager click commit button, permitted part timer can user blocked button

(3). If manager click cancel button, rejected part timer information is deleted in DB

**3. System Code**

**- JAVA CODE**

**1. Activity\_applyMySchedule.java**

package com.example.term;

import org.json.JSONArray;

import org.json.JSONObject;

import android.app.Activity;

import android.app.DatePickerDialog;

import android.app.Dialog;

import android.app.ProgressDialog;

import android.os.AsyncTask;

import android.os.Bundle;

import android.util.Log;

import android.view.View;

import android.view.View.OnClickListener;

import android.widget.Button;

import android.widget.CheckBox;

import android.widget.DatePicker;

import android.widget.TextView;

import android.widget.Toast;

//It supports to P-Timer when P-Timer applies P-Timer's task. So system receive P-Timer application.

public class Activity\_applyMySchedule extends Activity implements OnClickListener {

private static final int DIALOG\_DATE = 0;

private DBmanager db; //send query of P-Timer application and receive query

private TextView durationText; //show duration

private Button dateSelectBtn; //receive P-Timer input

private CheckBox openCheckBox; //receive P-Timer input

private CheckBox middleCheckBox; //receive P-Timer input

private CheckBox closeCheckBox; //receive P-Timer input

private Button saveBtn; //save user application

private String startYear, startMonth, startDays; //receive work start day of setting manager

private String endYear, endMonth, endDays; //receive work end day of setting manager

private String openNum, middleNum, closeNum;

private String isCheckOpen, isCheckMiddle, isCheckClose; //receive P-Timer application of each time

private String name, ID, locationCode, workDate; //get P-Timer information

//Initialize activity.

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_applymyschedule);

db = new DBmanager(); //reference DB

getActionBar().hide(); //hide actionbar

//get id

durationText = (TextView)findViewById(R.id.applyduration);

dateSelectBtn = (Button)findViewById(R.id.selectDataBtn);

openCheckBox = (CheckBox)findViewById(R.id.openCheckBox);

middleCheckBox = (CheckBox)findViewById(R.id.middleCheckBox);

closeCheckBox = (CheckBox)findViewById(R.id.closeCheckBox);

saveBtn = (Button)findViewById(R.id.saveBtn);

dateSelectBtn.setOnClickListener(this);

saveBtn.setOnClickListener(this);

ID = menuActivity.ID;

locationCode = menuActivity.locationCode;

name = menuActivity.name;

new getDateTask().execute(menuActivity.locationCode);

}

//get user input

@Override

public void onClick(View v) {

// TODO Auto-generated method stub

if(v.getId() == dateSelectBtn.getId())

showDialog(DIALOG\_DATE); //If user click calender, show calender that setting manager

else if(v.getId() == saveBtn.getId()) // if user click the save button, we check the all possible and put data to DB

{

isCheckOpen = Boolean.toString(openCheckBox.isChecked());

isCheckMiddle = Boolean.toString(middleCheckBox.isChecked());

isCheckClose = Boolean.toString(closeCheckBox.isChecked());

new saveWorkInfoTask().execute(locationCode,ID,name,workDate,isCheckOpen,isCheckMiddle,isCheckClose);

}

}

//Check checkbox clicked

public void onCheckBoxClicked(View v)

{

if(v.getId() == openCheckBox.getId())

{

if(openNum.equalsIgnoreCase("0"))

{

openCheckBox.setChecked(false);

Toast.makeText(this, "吏��썝 �씤�썝�씠 珥덇낵 �븯���뒿�땲�떎", Toast.LENGTH\_SHORT).show();

}

}

if(v.getId() == middleCheckBox.getId())

{

if(middleNum.equalsIgnoreCase("0"))

{

middleCheckBox.setChecked(false);

Toast.makeText(this, "吏��썝 �씤�썝�씠 珥덇낵 �븯���뒿�땲�떎", Toast.LENGTH\_SHORT).show();

}

}

if(v.getId() == middleCheckBox.getId())

{

Log.e("error","openNum : " + closeNum);

if(closeNum.equalsIgnoreCase("0"))

{

closeCheckBox.setChecked(false);

Toast.makeText(this, "吏��썝 �씤�썝�씠 珥덇낵 �븯���뒿�땲�떎", Toast.LENGTH\_SHORT).show();

}

}

}

//Show calender

protected Dialog onCreateDialog(int id) {

DatePickerDialog \_date = null;

switch (id) {

case DIALOG\_DATE:

\_date = new DatePickerDialog(this, dateListener,

Integer.parseInt(startYear), Integer.parseInt(startMonth)-1, Integer.parseInt(startDays)){

@Override

public void onDateChanged(DatePicker view, int year, int monthOfYear, int dayOfMonth)

{

//Limited part (Manager set date of work

if (year > Integer.parseInt(endYear))

view.updateDate(Integer.parseInt(endYear), Integer.parseInt(endMonth) - 1, Integer.parseInt(endDays));

if (monthOfYear > Integer.parseInt(endMonth) - 1 && year == Integer.parseInt(endYear))

view.updateDate(Integer.parseInt(endYear), Integer.parseInt(endMonth) - 1, Integer.parseInt(endDays));

if (dayOfMonth > Integer.parseInt(endDays) && year == Integer.parseInt(endYear) && monthOfYear == Integer.parseInt(endMonth) - 1)

view.updateDate(Integer.parseInt(endYear), Integer.parseInt(endMonth) - 1, Integer.parseInt(endDays));

//these below lines of code used for setting the maximum as well as minimum dates on Date Picker Dialog..

if (year < Integer.parseInt(startYear))

view.updateDate(Integer.parseInt(startYear), Integer.parseInt(startMonth) - 1, Integer.parseInt(startDays));

if (monthOfYear < Integer.parseInt(startMonth) - 1 && year == Integer.parseInt(startYear) )

view.updateDate(Integer.parseInt(startYear), Integer.parseInt(startMonth) - 1, Integer.parseInt(startDays));

if (dayOfMonth < Integer.parseInt(startDays) && year == Integer.parseInt(startYear) && monthOfYear == Integer.parseInt(startMonth) - 1)

view.updateDate(Integer.parseInt(startYear), Integer.parseInt(startMonth) - 1, Integer.parseInt(startDays));

}

};

}

return \_date;

}

//If user click the date, system get this and set string

private DatePickerDialog.OnDateSetListener dateListener = new DatePickerDialog.OnDateSetListener() {

@Override

public void onDateSet(DatePicker view, int year, int monthOfYear,

int dayOfMonth) {

dateSelectBtn.setText(year+"-"+(monthOfYear+1)+"-"+dayOfMonth);

openCheckBox.setChecked(false);

middleCheckBox.setChecked(false);

closeCheckBox.setChecked(false);

workDate = year+"-"+(monthOfYear+1)+"-"+dayOfMonth;

new getMySaveDataTask().execute(year+"-"+(monthOfYear+1)+ "-"+ dayOfMonth);

}

};

// If user click the calender, system get data setting manager

private class getMySaveDataTask extends AsyncTask<String, JSONArray, Void>

{

private final ProgressDialog dialog = new ProgressDialog(Activity\_applyMySchedule.this);

@Override

protected void onPreExecute() {

// TODO Auto-generated method stub

dialog.setMessage("���옣 �젙蹂대�� 遺덈윭�삤�뒗 以묒엯�땲�떎."); //show dialog to UI

dialog.show();

super.onPreExecute();

}

@Override

protected Void doInBackground(String... params) {

//set query

String query = "select applyTime from applytask\_info where ID = '" + ID + "' and workDate = '" + params[0] + "';";

//make JSONArray to use requesting query

JSONArray one = db.requestQuery(query);

publishProgress(one);

return null;

}

@Override

protected void onProgressUpdate(JSONArray... values) {

super.onProgressUpdate(values);

if(values[0].length() != 0){

try

{

JSONObject json\_data = values[0].getJSONObject(0); //get data and put the JSONObject

String applyTime = json\_data.getString("applyTime"); //get data

//set user state

if(applyTime.equals("123"))

{

openCheckBox.setChecked(true);

middleCheckBox.setChecked(true);

closeCheckBox.setChecked(true);

}

else if(applyTime.equals("12"))

{

openCheckBox.setChecked(true);

middleCheckBox.setChecked(false);

closeCheckBox.setChecked(true);

}

else if(applyTime.equals("23"))

{

openCheckBox.setChecked(false);

middleCheckBox.setChecked(true);

closeCheckBox.setChecked(true);

}

else if(applyTime.equals("13"))

{

openCheckBox.setChecked(true);

middleCheckBox.setChecked(false);

closeCheckBox.setChecked(true);

}

else if(applyTime.equals("1"))

{

openCheckBox.setChecked(true);

middleCheckBox.setChecked(false);

closeCheckBox.setChecked(false);

}

else if(applyTime.equals("2"))

{

openCheckBox.setChecked(false);

middleCheckBox.setChecked(true);

closeCheckBox.setChecked(false);

}

else if(applyTime.equals("3"))

{

openCheckBox.setChecked(false);

middleCheckBox.setChecked(false);

closeCheckBox.setChecked(true);

}

}catch(Exception e)

{

Log.e("error",e.getMessage());

}

}

}

@Override

protected void onPostExecute(Void result) {

if(dialog.isShowing())

dialog.dismiss(); //dismiss dialog

super.onPostExecute(result);

}

}

//save to database with user information of task application

private class saveWorkInfoTask extends AsyncTask<String, Void, Void>

{

private final ProgressDialog dialog = new ProgressDialog(Activity\_applyMySchedule.this);

@Override

protected void onPreExecute() {

// TODO Auto-generated method stub

dialog.setMessage("�옞�떆留� 湲곕떎�젮 二쇱꽭�슂."); //show dialog

dialog.show();

super.onPreExecute();

}

@Override

protected Void doInBackground(String... params) {

JSONArray one = db.requestQuery("select \* from applytask\_info where ID = '" + params[1] + "' and workDate = '" + params[3] + "';");

//request query

String query = null;

//check user work information and put the data of user work information to database

if(one.length() == 0)

{

if(params[4].equals("true") && params[5].equals("true") && params[6].equals("true"))

query = "insert into applytask\_info values('" + params[0] +"','" + params[1] + "','" + params[2] +"','" + params[3] + "','123'" + ")";

else if(params[4].equals("true") && params[5].equals("true") && params[6].equals("false"))

query = "insert into applytask\_info values('" + params[0] +"','" + params[1] + "','" + params[2] +"','" + params[3] + "','12'" + ")";

else if(params[4].equals("true") && params[5].equals("false") && params[6].equals("true"))

query = "insert into applytask\_info values('" + params[0] +"','" + params[1] + "','" + params[2] +"','" + params[3] + "','13'" + ")";

else if(params[4].equals("false") && params[5].equals("true") && params[6].equals("true"))

query = "insert into applytask\_info values('" + params[0] +"','" + params[1] + "','" + params[2] +"','" + params[3] + "','23'" + ")";

else if(params[4].equals("fasle") && params[5].equals("false") && params[6].equals("true"))

query = "insert into applytask\_info values('" + params[0] +"','" + params[1] + "','" + params[2] +"','" + params[3] + "','3'" + ")";

else if(params[4].equals("false") && params[5].equals("true") && params[6].equals("false"))

query = "insert into applytask\_info values('" + params[0] +"','" + params[1] + "','" + params[2] +"','" + params[3] + "','2'" + ")";

else if(params[4].equals("true") && params[5].equals("false") && params[6].equals("false"))

query = "insert into applytask\_info values('" + params[0] +"','" + params[1] + "','" + params[2] +"','" + params[3] + "','1'" + ")";

}

else

{

if(params[4].equals("true") && params[5].equals("true") && params[6].equals("true"))

query = "update location\_info set applyTime = '123';";

else if(params[4].equals("true") && params[5].equals("true") && params[6].equals("false"))

query = "update location\_info set applyTime = '12';";

else if(params[4].equals("true") && params[5].equals("false") && params[6].equals("true"))

query = "update location\_info set applyTime = '13';";

else if(params[4].equals("false") && params[5].equals("true") && params[6].equals("true"))

query = "update location\_info set applyTime = '23';";

else if(params[4].equals("fasle") && params[5].equals("false") && params[6].equals("true"))

query = "update location\_info set applyTime = '3';";

else if(params[4].equals("false") && params[5].equals("true") && params[6].equals("false"))

query = "update location\_info set applyTime = '2';";

else if(params[4].equals("true") && params[5].equals("false") && params[6].equals("false"))

query = "update location\_info set applyTime = '1';";

}

db.sendQuery(query); //send query

return null;

}

@Override

protected void onPostExecute(Void result) {

if(dialog.isShowing())

dialog.dismiss(); //dismiss dialog

Toast.makeText(getApplicationContext(), "�떊泥��씠 �셿猷� �릺�뿀�뒿�땲�떎." , Toast.LENGTH\_SHORT).show();

super.onPostExecute(result);

}

}

//get date information

private class getDateTask extends AsyncTask<String, JSONArray, Void>

{

private final ProgressDialog dialog = new ProgressDialog(Activity\_applyMySchedule.this);

@Override

protected void onPreExecute() {

// TODO Auto-generated method stub

dialog.setMessage("�옞�떆留� 湲곕떎�젮 二쇱꽭�슂."); // show dialog

dialog.show();

super.onPreExecute();

}

@Override

protected Void doInBackground(String... params) {

String query = "select startDate, endDate, openNum, middleNum, closeNum from location\_info where locationCode = '" + params[0] + "';";

JSONArray dateInfo = db.requestQuery(query); //request to database for getting data information of manager setting

publishProgress(dateInfo);

return null;

}

@Override

protected void onProgressUpdate(JSONArray... values) {

super.onProgressUpdate(values);

Log.e("error","+4");

try

{

JSONObject json\_data = values[0].getJSONObject(0);

//get data and set information

String startDate = json\_data.getString("startDate");

String endDate= json\_data.getString("endDate");

openNum = json\_data.getString("openNum");

middleNum = json\_data.getString("middleNum");

closeNum = json\_data.getString("closeNum");

String temp[] = startDate.split("-");

String temp2[] = endDate.split("-");

startYear = temp[0];

startMonth = temp[1];

startDays = temp[2];

endYear = temp2[0];

endMonth = temp2[1];

endDays = temp2[2];

durationText.setText("available apply duration : " + startDate + " ~ " + endDate);

}catch(Exception e)

{

Log.e("error",e.getMessage());

}

Log.e("error","+6");

}

@Override

protected void onPostExecute(Void result) {

if(dialog.isShowing())

dialog.dismiss(); //dismiss dialog

super.onPostExecute(result);

}

}

}

**2. Activity\_bulletin.java**

package com.example.term;

import android.app.Activity;

import android.os.Bundle;

import android.view.KeyEvent;

import android.view.View;

import android.view.View.OnClickListener;

import android.webkit.WebView;

import android.webkit.WebViewClient;

import android.widget.Button;

//This class supports bulletin board. if user click the bulletin button, system show web site that our bulletin board

public class Activity\_bulletin extends Activity implements OnClickListener{

private WebView mWebView; //using webview

private Button backButton; //get user action in web site

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_bulletin);

getActionBar().hide();

//set id to use java

backButton = (Button)findViewById(R.id.homeButton);

mWebView = (WebView)findViewById(R.id.webView);

backButton.setOnClickListener(this);

// set javascript enable

mWebView.getSettings().setJavaScriptEnabled(true);

// Show the this site that our bulletin board

mWebView.loadUrl("http://kyl1436.dothome.co.kr/xe/freeboard");

// intercept user action in web site

mWebView.setWebViewClient(new WebViewClientClass());

}

@Override

public void onBackPressed() {

// TODO Auto-generated method stub

mWebView.goBack(); //get user input and execute predictable action

}

@Override

public boolean onKeyDown(int keyCode, KeyEvent event) {

if ((keyCode == KeyEvent.KEYCODE\_BACK) && mWebView.canGoBack()) {

mWebView.goBack(); //get user input and execute predictable action

return true;

}

return super.onKeyDown(keyCode, event);

}

private class WebViewClientClass extends WebViewClient {

@Override

public boolean shouldOverrideUrlLoading(WebView view, String url) {

view.loadUrl(url); //show url

return true;

}

}

@Override

public void onClick(View v) {

finish(); //if user click home button, show our menu

}

}

**3. Activity\_enrollment.java**

**package** com.example.term;

**import** org.json.JSONArray;

**import** android.app.Activity;

**import** android.app.ProgressDialog;

**import** android.os.AsyncTask;

**import** android.os.Bundle;

**import** android.util.Log;

**import** android.view.View;

**import** android.view.View.OnClickListener;

**import** android.widget.Button;

**import** android.widget.EditText;

**import** android.widget.RadioButton;

**import** android.widget.RadioGroup;

**import** android.widget.Toast;

//if user want to join our system, we show this class

**public** **class** Activity\_enrollment **extends** Activity **implements** OnClickListener {

**private** EditText idEdit;

**private** EditText passEdit;

**private** EditText nameEdit;

**private** EditText locationEdit;

**private** EditText passReCheckEdit;

**private** RadioGroup radioGroup;

**private** RadioButton radioTimer;

**private** RadioButton radioManager;

**private** Button dupliBtn;

**private** Button okBtn;

**private** Button dupliLocationBtn;

**private** DBmanager db;

**private** **boolean** idCheck;

**private** **boolean** locationCheck;

String positionCode = **null**;

**private** String tempLocation;

//initialize this class.

**protected** **void** onCreate(Bundle savedInstanceState) {

**super**.onCreate(savedInstanceState);

setContentView(R.layout.***activity\_enrollment***);

getActionBar().hide(); //hide actionbar

//set id to use in java

idEdit = (EditText)findViewById(R.id.***idEditText***);

passEdit = (EditText)findViewById(R.id.***passEditText***);

nameEdit = (EditText)findViewById(R.id.***nameEditText***);

locationEdit = (EditText)findViewById(R.id.***locationEditText***);

passReCheckEdit = (EditText)findViewById(R.id.***passEditText2***);

dupliLocationBtn = (Button)findViewById(R.id.***shopCheckBtn***);

radioGroup = (RadioGroup)findViewById(R.id.***radioGroup***);

radioTimer = (RadioButton)findViewById(R.id.***radioBtnTimer***);

radioManager = (RadioButton)findViewById(R.id.***radioBtnManager***);

dupliBtn = (Button)findViewById(R.id.***idCheckButton***);

okBtn = (Button)findViewById(R.id.***OKButton9***);

//set db adapter

db = **new** DBmanager();

//set click listener

dupliBtn.setOnClickListener(**this**);

okBtn.setOnClickListener(**this**);

dupliLocationBtn.setOnClickListener(**this**);

}

@Override

**public** **void** onClick(View v) {

**if**(v.getId() == dupliBtn.getId()) //if user click the id check button, we send query to db and find duplication

{

String tempId = idEdit.getText().toString();

**if**(tempId.length() == 0)

Toast.*makeText*(**this**, "ID瑜� �엯�젰�빐 二쇱꽭�슂", Toast.***LENGTH\_SHORT***).show();

**else**

**new** idCheckTask().execute(idEdit.getText().toString()); //send query

}

**else** **if**(v.getId() == dupliLocationBtn.getId()) //if user click the location check button, we send query to db and find duplication

{

tempLocation = locationEdit.getText().toString();

**if**(radioTimer.isChecked() && locationEdit.getText().toString().length() != 0)

**new** locationCheckTask().execute("0",locationEdit.getText().toString());

**else** **if**(radioManager.isChecked()&& locationEdit.getText().toString().length() != 0)

**new** locationCheckTask().execute("1",locationEdit.getText().toString());

**else** **if**(radioManager.isChecked() == **false** && radioTimer.isChecked() == **false**)

Toast.*makeText*(**this**, "媛��엯�옄 type�쓣 �꽑�깮�빐 二쇱꽭�슂.", Toast.***LENGTH\_SHORT***).show();

**else**

Toast.*makeText*(**this**, "吏��뿭 code瑜� �엯�젰�빐 二쇱꽭�슂.", Toast.***LENGTH\_SHORT***).show();

}

**else** **if**(v.getId() == okBtn.getId()) //if user click ok button, system check all possible error such as id checking, location checking

{

**if**(idEdit.getText().toString().length() == 0 && passEdit.getText().toString().length() == 0 &&

nameEdit.length() == 0 && (radioTimer.isChecked() || radioManager.isChecked()))

Toast.*makeText*(**this**, "紐⑤뱺 �젙蹂대�� �엯�젰�빐二쇱꽭�슂", Toast.***LENGTH\_SHORT***).show();

**else** **if**(passEdit.getText().toString().equalsIgnoreCase(passReCheckEdit.getText().toString()) == **false**)

{

Toast.*makeText*(**this**, "鍮꾨�踰덊샇瑜� �솗�씤�빐 二쇱꽭�슂", Toast.***LENGTH\_SHORT***).show();

}

**else**

{

**if**(idCheck == **false**)

Toast.*makeText*(**this**, "ID 以묐났�솗�씤�쓣 �빐二쇱꽭�슂", Toast.***LENGTH\_SHORT***).show();

**else** **if**(locationCheck == **false**)

Toast.*makeText*(**this**, "吏��뿭Code 以묐났�솗�씤�쓣 �빐二쇱꽭�슂", Toast.***LENGTH\_SHORT***).show();

**else**

{

**if**(tempLocation.equals(locationEdit.getText().toString())){

**if**(radioTimer.isChecked())

positionCode = "0";

**else** **if**(radioManager.isChecked())

positionCode = "1";

String id = idEdit.getText().toString();

String pass = passEdit.getText().toString();

String name = nameEdit.getText().toString();

String locationCode = locationEdit.getText().toString();

String openCode = "0";

String query;

**if**(positionCode.equals("0"))

query = "insert into member\_info values('"+ id +"','" + pass + "','" + name + "','" + locationCode + "','" + positionCode + "','" + openCode +"');";

**else**

query = "insert into member\_info values('"+ id +"','" + pass + "','" + name + "','" + locationCode + "','" + positionCode + "','" + openCode +"','" + 1 +"');";

**new** applyEnrollmentTask().execute(query,locationCode, positionCode);

}

**else**

Toast.*makeText*(**this**, "吏��뿭Code 以묐났�솗�씤�쓣 �빐二쇱꽭�슂", Toast.***LENGTH\_SHORT***).show();

}

}

}

}

//If system check all possible error and it can join our system, put user information to db and approve user

**private** **class** applyEnrollmentTask **extends** AsyncTask<String, Boolean,Void>

{

**private** **final** ProgressDialog dialog = **new** ProgressDialog(Activity\_enrollment.**this**);

@Override

**protected** **void** onPreExecute() {

// **TODO** Auto-generated method stub

dialog.setMessage("�옞�떆留� 湲곕떎�젮 二쇱꽭�슂.");

dialog.show(); //show dialog

**super**.onPreExecute();

}

@Override

**protected** Void doInBackground(String... params) {

**boolean** check1 = **false**;

**boolean** check2 = **false**;

**if**(params[2].equalsIgnoreCase("1"))

{

String query = "insert into location\_info values('" + params[1] + "','0','0','0','0','0','0');";

check2 = db.sendQuery(query);

}

check1 = db.sendQuery(params[0]);

**if**(positionCode.equals("1")){

check1 = !check1;

check2 = !check2;

}

publishProgress(!(check2 && check1));

**return** **null**;

}

@Override

**protected** **void** onProgressUpdate(Boolean... values) {

**if**(values[0] == **true**)

{

Toast.*makeText*(getApplicationContext(), "媛��엯�씠 �셿猷뚮릺�뿀�뒿�땲�떎", Toast.***LENGTH\_SHORT***).show();

finish();

}

**else**

{

Toast.*makeText*(getApplicationContext(), "�떎�떆 �떆�룄�빐 二쇱꽭�슂", Toast.***LENGTH\_SHORT***).show();

}

**super**.onProgressUpdate(values);

}

@Override

**protected** **void** onPostExecute(Void result) {

**if**(dialog.isShowing())

dialog.dismiss();

**super**.onPostExecute(result);

}

}

//id check test, request to db and get data and check duplication

**private** **class** idCheckTask **extends** AsyncTask<String, Integer, Void>{

**private** **final** ProgressDialog dialog = **new** ProgressDialog(Activity\_enrollment.**this**);

@Override

**protected** **void** onPreExecute() {

// **TODO** Auto-generated method stub

dialog.setMessage("�옞�떆留� 湲곕떎�젮 二쇱꽭�슂.");

dialog.show();

**super**.onPreExecute();

}

@Override

**protected** Void doInBackground(String... params) {

JSONArray idArray = db.requestQuery("select ID from member\_info where ID ='" + params[0] +"';");

publishProgress(idArray.length());

**return** **null**;

}

@Override

**protected** **void** onProgressUpdate(Integer... values) {

**if**(values[0] == 0)

{

Toast.*makeText*(getApplicationContext(), "�궗�슜�븷�닔 �엳�뒗 ID �엯�땲�떎", Toast.***LENGTH\_SHORT***).show();

idCheck = **true**;

}

**else**

Toast.*makeText*(getApplicationContext(), "以묐났�릺�뒗 ID �엯�땲�떎", Toast.***LENGTH\_SHORT***).show();

**super**.onProgressUpdate(values);

}

@Override

**protected** **void** onPostExecute(Void result) {

**if**(dialog.isShowing())

dialog.dismiss();

**super**.onPostExecute(result);

}

}

//location check test, request to db and get data and check duplication

**private** **class** locationCheckTask **extends** AsyncTask<String, String, Void>{

**private** **final** ProgressDialog dialog = **new** ProgressDialog(Activity\_enrollment.**this**);

**private** **int** arrLength;

@Override

**protected** **void** onPreExecute() {

// **TODO** Auto-generated method stub

dialog.setMessage("�옞�떆留� 湲곕떎�젮 二쇱꽭�슂.");

dialog.show();

**super**.onPreExecute();

}

@Override

**protected** Void doInBackground(String... params) {

JSONArray idArray = db.requestQuery("select locationCode from location\_info where locationCode = '" + params[1] + "';");

Log.*e*("error","type : " + params[0]);

Log.*e*("error","code : " + params[1]);

Log.*e*("error","length : " + idArray.length());

arrLength = idArray.length();

publishProgress(params[0]);

**return** **null**;

}

@Override

**protected** **void** onProgressUpdate(String... values) {

// **TODO** Auto-generated method stub

**if**(values[0].equalsIgnoreCase("0") && arrLength > 0)

{

Toast.*makeText*(getApplicationContext(), "�궗�슜 媛��뒫�븳 Code �엯�땲�떎", Toast.***LENGTH\_LONG***).show();

locationCheck = **true**;

}

**else** **if**(values[0].equalsIgnoreCase("0") && arrLength == 0)

Toast.*makeText*(getApplicationContext(), "議댁옱�븯吏� �븡�뒗 Code �엯�땲�떎", Toast.***LENGTH\_LONG***).show();

**else** **if**(values[0].equalsIgnoreCase("1") && arrLength == 0)

{

Toast.*makeText*(getApplicationContext(), "�궗�슜 媛��뒫�븳 Code �엯�땲�떎", Toast.***LENGTH\_LONG***).show();

locationCheck = **true**;

}

**else** **if**(values[0].equalsIgnoreCase("1") && arrLength > 0)

Toast.*makeText*(getApplicationContext(), "以묐났�릺�뒗 Code �엯�땲�떎", Toast.***LENGTH\_LONG***).show();

**super**.onProgressUpdate(values);

}

@Override

**protected** **void** onPostExecute(Void result) {

**if**(dialog.isShowing())

dialog.dismiss();

**super**.onPostExecute(result);

}

}

}

**4. Activity\_fixSchedule.java**

package com.example.term;

import java.util.ArrayList;

import java.util.Date;

import org.json.JSONArray;

import org.json.JSONObject;

import android.app.Activity;

import android.app.DatePickerDialog;

import android.app.Dialog;

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.View.OnClickListener;

import android.widget.ArrayAdapter;

import android.widget.Button;

import android.widget.DatePicker;

import android.widget.ListView;

import android.widget.TextView;

//this class supports manager if manager want to fix schedule, show this class

public class Activity\_fixSchedule extends Activity implements OnClickListener{

private Button catalBtn, showFixMemBtn;

//using listview

private static ArrayList<String> applicantOpen;

private static ArrayList<String> applicantMiddle;

private static ArrayList<String> applicantClose;

//show each time list person

private ListView applicantOpenView;

private ListView applicantMiddleView;

private ListView applicantCloseView;

private TextView openText;

private TextView middleText;

private TextView closeText;

private TextView dateText;

//using listview

private ArrayAdapter<String> openAdapter;

private ArrayAdapter<String> middleAdapter;

private ArrayAdapter<String> closeAdapter;

public static String date;

private final int DIALOG\_DATE = 1;

private DBmanager db; //set database manager

@Override

protected void onCreate(Bundle savedInstanceState) {

setContentView(R.layout.activity\_fixschedule);

getActionBar().hide();

//set view id to use in java

catalBtn = (Button)findViewById(R.id.fixScedule\_showCatalogBtn);

showFixMemBtn = (Button)findViewById(R.id.fixScedule\_showfixmemberList);

applicantOpenView = (ListView)findViewById(R.id.fixScedule\_open);

applicantMiddleView = (ListView)findViewById(R.id.fixScedule\_middle);

applicantCloseView = (ListView)findViewById(R.id.fixScedule\_close);

openText = (TextView)findViewById(R.id.fix\_textView1);

middleText = (TextView)findViewById(R.id.fix\_textView2);

closeText = (TextView)findViewById(R.id.fix\_textView3);

dateText = (TextView)findViewById(R.id.fixSchedule\_dateText);

db = new DBmanager(); //get DB reference

applicantOpen = new ArrayList<String>();

applicantMiddle = new ArrayList<String>();

applicantClose = new ArrayList<String>();

openAdapter = new ArrayAdapter<String>(this,android.R.layout.simple\_list\_item\_1,applicantOpen);

middleAdapter = new ArrayAdapter<String>(this,android.R.layout.simple\_list\_item\_1,applicantMiddle);

closeAdapter = new ArrayAdapter<String>(this,android.R.layout.simple\_list\_item\_1,applicantClose);

applicantOpenView.setAdapter(openAdapter);

applicantMiddleView.setAdapter(middleAdapter);

applicantCloseView.setAdapter(closeAdapter);

catalBtn.setOnClickListener(this);

showFixMemBtn.setOnClickListener(this);

super.onCreate(savedInstanceState);

}

@Override

public void onClick(View v) {

switch(v.getId()){

case R.id.fixScedule\_showCatalogBtn: //if manager click the calender then show calender.

showDialog(DIALOG\_DATE);

break;

case R.id.fixScedule\_showfixmemberList: //if manager want random fix button, show rotation class

startActivityForResult(new Intent(this, Activity\_rotationApplicant.class), 100);

}

}

@Override

protected void onActivityResult(int requestCode, int resultCode, Intent data) {

// TODO Auto-generated method stub

super.onActivityResult(requestCode, resultCode, data);

}

@Override

protected Dialog onCreateDialog(int id) {

Date one = new Date();

if(id == DIALOG\_DATE) //show calender

return new DatePickerDialog(this, dateListener,2015,5,1);

return null;

}

private DatePickerDialog.OnDateSetListener dateListener =

new DatePickerDialog.OnDateSetListener() {

@Override

public void onDateSet(DatePicker view, int year, int monthOfYear,

int dayOfMonth) {

//if user click date, get this date

date = year+"-"+(monthOfYear+1)+"-"+dayOfMonth;

dateText.setText(date);

new getApplyMember().execute(menuActivity.locationCode,date);

}

};

// if manager click the date, set the information of P-Timer application

private class getApplyMember extends AsyncTask<String, JSONArray, Void>{

private final ProgressDialog dialog = new ProgressDialog(Activity\_fixSchedule.this);

@Override

protected void onPreExecute() {

// TODO Auto-generated method stub

dialog.setMessage("�옞�떆留� 湲곕떎�젮 二쇱꽭�슂."); //show dialog

dialog.show();

super.onPreExecute();

}

@Override

protected Void doInBackground(String... params) {

String locationCode = params[0];

String workDate = params[1];

String query = "select ID, name, applyTime from applytask\_info where locationCode = '" + locationCode + "' and workDate = '" + workDate +"';";

//send query and get data

JSONArray one = db.requestQuery(query);

publishProgress(one);

return null;

}

@Override

protected void onProgressUpdate(JSONArray... values) {

// TODO Auto-generated method stub

try

{ //clear listview

applicantOpen.clear();

applicantMiddle.clear();

applicantClose.clear();

for(int i=0; i<values[0].length(); i++)

{ //get P-Timer that apply this date

JSONObject json\_data = values[0].getJSONObject(i);

String ID = json\_data.getString("ID");

String name = json\_data.getString("name");

String applyTime = json\_data.getString("applyTime");

if(applyTime.contains("1"))

applicantOpen.add("ID : " + ID + "Name : " + name);

if(applyTime.contains("2"))

applicantMiddle.add("ID : " + ID + "Name : " + name);

if(applyTime.contains("3"))

applicantClose.add("ID : " + ID + "Name : " + name);

}

openText.setText("Open Time applyer : " + applicantOpen.size());

middleText.setText("Middle Time applyer: " + applicantMiddle.size());

closeText.setText("Close Time applyer : " + applicantClose.size());

openAdapter.notifyDataSetChanged();

middleAdapter.notifyDataSetChanged();

closeAdapter.notifyDataSetChanged();

}catch(Exception e)

{

Log.e("error",e.getMessage());

}

super.onProgressUpdate(values);

}

@Override

protected void onPostExecute(Void result) {

if(dialog.isShowing())

dialog.dismiss(); //dismiss dialog

super.onPostExecute(result);

}

}

}

**5. Activity\_newPeopleManage.java**

package com.example.term;

import org.json.JSONArray;

import org.json.JSONObject;

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.View.OnClickListener;

import android.widget.AdapterView;

import android.widget.AdapterView.OnItemClickListener;

import android.widget.Button;

import android.widget.ListView;

import android.widget.Toast;

//This class supports to manager. it can make that manager to manage location P-Timer

public class Activity\_newPeopleManage extends Activity implements OnClickListener, OnItemClickListener{

private ListView myListView; //show P-Timer state

private CustomAdapter myAdapter;

private DBmanager db; //get data through this DBmanager

private int pos;

@Override

protected void onCreate(Bundle savedInstanceState) {

// TODO Auto-generated method stub

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_newpeoplemanage);

getActionBar().hide(); //hide actionbar

//set P-Timer to listview

myListView = (ListView)findViewById(R.id.peopleListView);

myListView.setOnItemClickListener(this);

myAdapter = new CustomAdapter(this); //set adapter

myListView.setAdapter(myAdapter);

db = new DBmanager(); //set DB manager

new getMyPeopleTask().execute(); //get P-Timer information that in same location

}

//get P-Timer information and set listview

private class getMyPeopleTask extends AsyncTask<String, JSONArray, Void>

{

private final ProgressDialog dialog = new ProgressDialog(Activity\_newPeopleManage.this);

@Override

protected void onPreExecute() {

// TODO Auto-generated method stub

dialog.setMessage("�옞�떆留� 湲곕떎�젮 二쇱꽭�슂.");

dialog.show(); //show dialog

super.onPreExecute();

}

@Override

protected Void doInBackground(String... params) {

//set location code using request query

String myLocationCode = menuActivity.locationCode;

String query = "select ID, name, openCode from member\_info where locationCode = '" + myLocationCode + "' and positionCode = '0';";

//set JSONArray to request query

JSONArray myPeopleArr = db.requestQuery(query);

publishProgress(myPeopleArr);

return null;

}

@Override

protected void onProgressUpdate(JSONArray... values) {

try

{

for(int i=0; i<values[0].length(); i++)

{

JSONObject json\_data = values[0].getJSONObject(i);

//get P-Timer information

String ID = json\_data.getString("ID");

String name = json\_data.getString("name");

String openCode = json\_data.getString("openCode");

myAdapter.add(ID +" " +name + " " + openCode);

}

}catch(Exception e)

{

Log.e("error",e.getMessage());

}

super.onProgressUpdate(values);

}

@Override

protected void onPostExecute(Void result) {

if(dialog.isShowing())

dialog.dismiss(); // dismiss dialog

super.onPostExecute(result);

}

}

//if manager click accept button, we reset P-Timer information in DB

private class acceptMemberTask extends AsyncTask<String, Void, Void>

{

DBmanager db = new DBmanager(); //Set DB manager

private final ProgressDialog dialog = new ProgressDialog(Activity\_newPeopleManage.this);

@Override

protected void onPreExecute() {

// TODO Auto-generated method stub

dialog.setMessage("�옞�떆留� 湲곕떎�젮 二쇱꽭�슂.");

dialog.show(); //show dialog

super.onPreExecute();

}

@Override

protected Void doInBackground(String... params) {

String[] result = params[0].split(" ");

String ID = result[0];

String name = result[1];

//make query and update database

String query = "update member\_info set openCode = '1' where ID = '" + ID + "' and name = '" + name + "';";

if(db.sendQuery(query))

Log.e("error","荑쇰━ �옒蹂대깂");

else

{

Log.e("error","荑쇰━ �옒紐삳낫�깂");

}

return null;

}

@Override

protected void onProgressUpdate(Void... values) {

// TODO Auto-generated method stub

super.onProgressUpdate(values);

}

@Override

protected void onPostExecute(Void result) {

if(dialog.isShowing())

dialog.dismiss(); //dismiss dialog

Toast.makeText(getApplicationContext(), "�닔�씫 �릺�뿀�뒿�땲�떎" , Toast.LENGTH\_SHORT).show();

super.onPostExecute(result);

}

}

//if manager click the reject button, we reset P-Timer information in DB

private class rejectMemberTask extends AsyncTask<String, Void, Void>

{

DBmanager db = new DBmanager();

private final ProgressDialog dialog = new ProgressDialog(Activity\_newPeopleManage.this);

@Override

protected void onPreExecute() {

// TODO Auto-generated method stub

dialog.setMessage("�옞�떆留� 湲곕떎�젮 二쇱꽭�슂.");

dialog.show();

super.onPreExecute();

}

@Override

protected Void doInBackground(String... params) {

String[] result = params[0].split(" ");

String ID = result[0];

String name = result[1];

String query = "delete from member\_info where ID = '" + ID + "' and name = '" + name +"';";

if(db.sendQuery(query))

Log.e("error","荑쇰━ �옒蹂대깂dd");

else

{

Log.e("error","荑쇰━ �옒紐삳낫�깂dd");

}

return null;

}

@Override

protected void onPostExecute(Void result) {

if(dialog.isShowing())

dialog.dismiss();

Toast.makeText(getApplicationContext(), "�궘�젣 �릺�뿀�뒿�땲�떎" , Toast.LENGTH\_SHORT).show();

super.onPostExecute(result);

}

}

@Override

public void onClick(View v) {

if(v.getId() == R.id.acceptButton)

{

int pos = (Integer)v.getTag();

new acceptMemberTask().execute((String)myAdapter.getItem((pos)));

Button btn = (Button)findViewById(R.id.acceptButton);

btn.setEnabled(false);

}

if(v.getId() == R.id.rejectButton)

{

int pos = (Integer)v.getTag();

new rejectMemberTask().execute((String)myAdapter.getItem(pos));

myAdapter.remove(pos);

myAdapter.notifyDataSetChanged();

}

}

@Override

public void onItemClick(AdapterView<?> parent, View view, int position,

long id) {

pos = position;

// TODO Auto-generated method stub

}

}

**6. Activity\_rotationApplicant.java**

package com.example.term;

import java.util.ArrayList;

import org.json.JSONArray;

import org.json.JSONException;

import org.json.JSONObject;

import android.app.Activity;

import android.app.ProgressDialog;

import android.os.AsyncTask;

import android.os.Bundle;

import android.util.Log;

import android.view.View;

import android.view.View.OnClickListener;

import android.widget.ArrayAdapter;

import android.widget.Button;

import android.widget.ListView;

//if manager want random rotation, we supports through this class

public class Activity\_rotationApplicant extends Activity implements OnClickListener{

private Button rotBtn;

private Button okBtn;

private ListView openList;

private ListView middleList;

private ListView closeList;

//get user input and remove or plus to each time

private ArrayList<ApplyPerson> applyArr;

//set list

private ArrayList<String> openApplyArr;

private ArrayList<String> middleApplyArr;

private ArrayList<String> closeApplyArr;

//set list adapter

private ArrayAdapter<String> openAdapter;

private ArrayAdapter<String> middleAdapter;

private ArrayAdapter<String> closeAdapter;

private DBmanager db;

// initialize this class

@Override

protected void onCreate(Bundle savedInstanceState) {

// TODO Auto-generated method stub

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_rotationapplicant);

getActionBar().hide(); //hid actionbar

//set id to use in java

rotBtn = (Button)findViewById(R.id.rotation\_roBtn);

okBtn = (Button)findViewById(R.id.rotation\_okBtn);

openList = (ListView)findViewById(R.id.rotation\_opneList);

middleList = (ListView)findViewById(R.id.rotation\_middleList);

closeList = (ListView)findViewById(R.id.rotation\_closeList);

//set click listener

rotBtn.setOnClickListener(this);

okBtn.setOnClickListener(this);

db = new DBmanager();

applyArr = new ArrayList<ApplyPerson>();

openApplyArr = new ArrayList<String>();

middleApplyArr = new ArrayList<String>();

closeApplyArr = new ArrayList<String>();

openAdapter = new ArrayAdapter<String>(this, android.R.layout.simple\_spinner\_dropdown\_item,openApplyArr );

middleAdapter = new ArrayAdapter<String>(this, android.R.layout.simple\_spinner\_dropdown\_item,middleApplyArr );

closeAdapter = new ArrayAdapter<String>(this, android.R.layout.simple\_spinner\_dropdown\_item,closeApplyArr );

openList.setAdapter(openAdapter);

middleList.setAdapter(middleAdapter);

closeList.setAdapter(closeAdapter);

}

@Override

public void onClick(View v) {

if(v.getId() == rotBtn.getId()){ //if manager click rotation button, system rotate list randomly

//clear list

openApplyArr.clear();

middleApplyArr.clear();

closeApplyArr .clear();

new getApplyPersonTask().execute();

}

if(v.getId() == okBtn.getId()){

new pushApplyPersonTask().execute(); //if user click the ok button, set database

this.finish();

}

}

//if manager ok button, we set this information to database

private class pushApplyPersonTask extends AsyncTask<String, JSONArray, Void>{

private final ProgressDialog dialog = new ProgressDialog(Activity\_rotationApplicant.this);

@Override

protected void onPreExecute() {

// TODO Auto-generated method stub

dialog.setMessage("占쎌삛占쎈뻻筌랃옙 疫꿸퀡�뼄占쎌젻 雅뚯눘苑�占쎌뒄."); //show dialog

dialog.show();

super.onPreExecute();

}

@Override

protected Void doInBackground(String... params) {

//insert to database

for(int i=0; i<openApplyArr.size(); i++){

String query = "insert into fixed\_info values ('"+menuActivity.locationCode+"','"+openApplyArr.get(i)+"','"+Activity\_fixSchedule.date+"', '"+1+"');" ;

db.sendQuery(query);

}

for(int i=0; i<middleApplyArr.size(); i++){

String query = "insert into fixed\_info values ('"+menuActivity.locationCode+"','"+middleApplyArr.get(i)+"','"+Activity\_fixSchedule.date+"', '"+2+"');" ;

db.sendQuery(query);

}

for(int i=0; i<closeApplyArr.size(); i++){

String query = "insert into fixed\_info values ('"+menuActivity.locationCode+"','"+closeApplyArr.get(i)+"','"+Activity\_fixSchedule.date+"', '"+3+"');" ;

db.sendQuery(query);

}

publishProgress();

return null;

}

@Override

protected void onProgressUpdate(JSONArray... values) {

super.onProgressUpdate(values);

}

@Override

protected void onPostExecute(Void result) {

if(dialog.isShowing())

dialog.dismiss(); //dismiss dialog

super.onPostExecute(result);

}

}

//get P-Timer data in this date and rotate randomly.

private class getApplyPersonTask extends AsyncTask<String, JSONArray, Void>{

private final ProgressDialog dialog = new ProgressDialog(Activity\_rotationApplicant.this);

String openNum;

String middleNum;

String closeNum;

@Override

protected void onPreExecute() {

// TODO Auto-generated method stub

dialog.setMessage("�옞�떆留� 湲곕떎�젮 二쇱꽭�슂."); //show dialog

dialog.show();

super.onPreExecute();

}

@Override

protected Void doInBackground(String... params) {

String query = "select ID, name, applyTime from applytask\_info where locationCode = '" + menuActivity.locationCode +"' and workDate = '" + Activity\_fixSchedule.date + "';";

JSONArray one = db.requestQuery(query);

//send query to receive each time number information

String query2 = "select openNum, middleNum, closeNum from location\_info where locationCode = '" + menuActivity.locationCode +"';";

JSONArray one2 = db.requestQuery(query2);

JSONObject json\_data2;

//send query and get P-Timer information in this date

try {

json\_data2 = one2.getJSONObject(0);

openNum = json\_data2.getString("openNum");

middleNum = json\_data2.getString("middleNum");

closeNum = json\_data2.getString("closeNum");

} catch (JSONException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

publishProgress(one);

return null;

}

@Override

protected void onProgressUpdate(JSONArray... values) {

try

{

applyArr.clear();

//get each P-Timer information of application this date

for(int i=0; i<values[0].length(); i++){

JSONObject json\_data = values[0].getJSONObject(i);

String ID = json\_data.getString("ID");

String name= json\_data.getString("name");

String applyTime = json\_data.getString("applyTime");

applyArr.add(new ApplyPerson(ID,name,applyTime));

}

//set max number

int open = Integer.parseInt(openNum);

int middle = Integer.parseInt(middleNum);

int close = Integer.parseInt(closeNum);

int checkOpen, checkMiddle, checkClose;

checkOpen = checkMiddle = checkClose = 0;

int getSize = applyArr.size();

int changeSize = applyArr.size();

//Our random algorithm.

for(int i=0; i<getSize; i++){

int random = 0;

random = (int)(Math.random() \*1000) % changeSize;

if( checkOpen == open && checkMiddle == middle & checkClose == close){

break;

}

//if P-Timer apply '123', '12' ... (each state), system randomly set list only one

if(applyArr.get(random).getApplyTime().equals("123")){

if(checkOpen < open && checkMiddle < middle && checkClose < close){

int rnd = (int)(Math.random() \* 10) % 3;

if(rnd == 0){

checkOpen++;

openApplyArr.add(applyArr.get(random).getName());

applyArr.remove(random);

changeSize--;

}else if(rnd == 1){

checkMiddle++;

middleApplyArr.add(applyArr.get(random).getName());

applyArr.remove(random);

changeSize--;

}else if(rnd == 2){

checkClose++;

closeApplyArr.add(applyArr.get(random).getName());

applyArr.remove(random);

changeSize--;

}

}else if(checkOpen >= open && checkMiddle < middle && checkClose < close){

int rnd = (int)(Math.random() \* 10) % 2;

if(rnd == 0){

checkMiddle++;

middleApplyArr.add(applyArr.get(random).getName());

applyArr.remove(random);

changeSize--;

}else if(rnd == 1){

checkClose++;

closeApplyArr.add(applyArr.get(random).getName());

applyArr.remove(random);

changeSize--;

}

}else if(checkOpen < open && checkMiddle >= middle && checkClose < close){

int rnd = (int)(Math.random() \* 10) % 2;

if(rnd == 0){

checkOpen++;

openApplyArr.add(applyArr.get(random).getName());

applyArr.remove(random);

changeSize--;

}else if(rnd == 1){

checkClose++;

closeApplyArr.add(applyArr.get(random).getName());

applyArr.remove(random);

changeSize--;

}

}else if(checkOpen < open && checkMiddle < middle && checkClose >= close){

int rnd = (int)(Math.random() \* 10) % 2;

if(rnd == 0){

checkOpen++;

openApplyArr.add(applyArr.get(random).getName());

applyArr.remove(random);

changeSize--;

}else if(rnd == 1){

checkMiddle++;

middleApplyArr.add(applyArr.get(random).getName());

applyArr.remove(random);

changeSize--;

}

}else if(checkOpen < open && checkMiddle >= middle && checkClose >= close){

checkOpen++;

openApplyArr.add(applyArr.get(random).getName());

applyArr.remove(random);

changeSize--;

}else if(checkOpen >= open && checkMiddle < middle && checkClose >= close){

checkMiddle++;

middleApplyArr.add(applyArr.get(random).getName());

applyArr.remove(random);

changeSize--;

}else if(checkOpen >= open && checkMiddle >= middle && checkClose < close){

checkClose++;

closeApplyArr.add(applyArr.get(random).getName());

applyArr.remove(random);

changeSize--;

}

//--------------------------------------------

}else if(applyArr.get(random).getApplyTime().equals("13")){

if(checkOpen < open && checkClose < close){

int rnd = (int)(Math.random() \* 10) % 2;

if(rnd == 0){

checkOpen++;

openApplyArr.add(applyArr.get(random).getName());

applyArr.remove(random);

changeSize--;

}else{

checkClose++;

closeApplyArr.add(applyArr.get(random).getName());

applyArr.remove(random);

changeSize--;

}

}else if(checkOpen >= open && checkClose < close){

checkClose++;

closeApplyArr.add(applyArr.get(random).getName());

applyArr.remove(random);

changeSize--;

}else if(checkOpen < open && checkClose >= close){

checkOpen++;

openApplyArr.add(applyArr.get(random).getName());

applyArr.remove(random);

changeSize--;

}

//-----------------------------------------------

}else if(applyArr.get(random).getApplyTime().equals("12")){

if(checkOpen < open && checkMiddle < middle){

int rnd = (int)(Math.random() \* 10) % 2;

if(rnd == 0){

checkOpen++;

openApplyArr.add(applyArr.get(random).getName());

applyArr.remove(random);

changeSize--;

}else{

checkMiddle++;

middleApplyArr.add(applyArr.get(random).getName());

applyArr.remove(random);

changeSize--;

}

}else if(checkOpen < open && checkMiddle >= middle){

checkOpen++;

openApplyArr.add(applyArr.get(random).getName());

applyArr.remove(random);

changeSize--;

}else if(checkOpen >= open && checkMiddle < middle){

checkMiddle++;

middleApplyArr.add(applyArr.get(random).getName());

applyArr.remove(random);

changeSize--;

}

}else if(applyArr.get(random).getApplyTime().equals("1")){

if(checkOpen < open){

checkOpen++;

openApplyArr.add(applyArr.get(random).getName());

applyArr.remove(random);

changeSize--;

}

}else if(applyArr.get(random).getApplyTime().equals("2")){

if(checkMiddle < middle){

checkMiddle++;

middleApplyArr.add(applyArr.get(random).getName());

applyArr.remove(random);

changeSize--;

}

}else if(applyArr.get(random).getApplyTime().equals("3")){

if(checkClose < close){

checkClose++;

closeApplyArr.add(applyArr.get(random).getName());

applyArr.remove(random);

changeSize--;

}

}else if(applyArr.get(random).getApplyTime().equals("23")){

if(checkMiddle < middle && checkClose < close){

int rnd = (int)(Math.random() \* 10) % 2;

if(rnd == 0){

checkMiddle++;

middleApplyArr.add(applyArr.get(random).getName());

applyArr.remove(random);

changeSize--;

}else{

checkClose++;

closeApplyArr.add(applyArr.get(random).getName());

applyArr.remove(random);

changeSize--;

}

}else if(checkMiddle < middle && checkClose >= close){

checkMiddle++;

middleApplyArr.add(applyArr.get(random).getName());

applyArr.remove(random);

changeSize--;

}else if(checkMiddle >= middle && checkClose < close){

checkClose++;

closeApplyArr.add(applyArr.get(random).getName());

applyArr.remove(random);

changeSize--;

}

}

}

}catch(Exception e){

Log.e("error",e.getMessage());

}

super.onProgressUpdate(values);

}

@Override

protected void onPostExecute(Void result) {

if(dialog.isShowing())

dialog.dismiss(); //dismiss dialog

openAdapter.notifyDataSetChanged();

middleAdapter.notifyDataSetChanged();

closeAdapter.notifyDataSetChanged();

super.onPostExecute(result);

}

}

}

**7. Activity\_scheduleManage.java**

package com.example.term;

import org.json.JSONArray;

import android.app.AlertDialog;

import android.content.DialogInterface;

import android.app.Activity;

import android.app.DatePickerDialog;

import android.app.Dialog;

import android.app.ProgressDialog;

import android.os.AsyncTask;

import android.os.Bundle;

import android.util.Log;

import android.view.View;

import android.view.View.OnClickListener;

import android.widget.AdapterView;

import android.widget.AdapterView.OnItemSelectedListener;

import android.widget.ArrayAdapter;

import android.widget.Button;

import android.widget.DatePicker;

import android.widget.Spinner;

import android.widget.TextView;

import android.widget.Toast;

public class Activity\_scheduleManage extends Activity implements OnClickListener{

private static final int DIALOG\_DATE = 0;

private static final int DIALOG\_DATE2 = 2;

private DBmanager db;

private TextView openText, middleText, closeText;

private Button commitButton;

Button datePickButton,dateWorkButton ;

private String possibleDate = "", workDate = "";

private String openN, middleN, closeN;

private String[] ch1 = {"1","2","3","4","5","6","7" }; //set number of each time

private String[] ch2 = {"1","2","3","4","5","6","7","8","9"};

private String[] ch3 = {"1","2","3","4","5","6","7","8"};

private Spinner open, middle, close;

@Override

public void onCreate(Bundle savedInstanceState){

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_schedulemanage);

getActionBar().hide(); //hide actionbar

//set id to use in java

datePickButton = (Button)findViewById(R.id.datePickButton);

dateWorkButton = (Button)findViewById(R.id.datePickButton2);

open = (Spinner)findViewById(R.id.openNumber);

middle = (Spinner)findViewById(R.id.middleNumber);

close = (Spinner)findViewById(R.id.closeNumber);

open.setPrompt("Open ");

middle.setPrompt("Middle ");

close.setPrompt("Close ");

//set adapter

ArrayAdapter<String> list1 = new ArrayAdapter<String>(this, android.R.layout.simple\_spinner\_dropdown\_item,ch1 );

open.setAdapter(list1);

ArrayAdapter<String> list2 = new ArrayAdapter<String>(this, android.R.layout.simple\_spinner\_dropdown\_item,ch2 );

middle.setAdapter(list2);

ArrayAdapter<String> list3 = new ArrayAdapter<String>(this, android.R.layout.simple\_spinner\_dropdown\_item,ch3 );

close.setAdapter(list3);

Log.e("error","4");

datePickButton.setOnClickListener(this);

dateWorkButton.setOnClickListener(this);

Log.e("error","5");

//get Number of people each time

open.setOnItemSelectedListener(new OnItemSelectedListener(){

@Override

public void onItemSelected(AdapterView<?> parent, View view,

int position, long id) {

// TODO Auto-generated method stub

//selMenu = (String) menuSpinner.getSelectedItem();

openN = (String)open.getSelectedItem();

}

@Override

public void onNothingSelected(AdapterView<?> parent) {

// TODO Auto-generated method stub

}

});

middle.setOnItemSelectedListener(new OnItemSelectedListener(){

@Override

public void onItemSelected(AdapterView<?> parent, View view,

int position, long id) {

// TODO Auto-generated method stub

middleN = (String)middle.getSelectedItem();

}

@Override

public void onNothingSelected(AdapterView<?> parent) {

// TODO Auto-generated method stub

}

});

close.setOnItemSelectedListener(new OnItemSelectedListener(){

@Override

public void onItemSelected(AdapterView<?> parent, View view,

int position, long id) {

// TODO Auto-generated method stub

closeN = (String)close.getSelectedItem();

}

@Override

public void onNothingSelected(AdapterView<?> parent) {

// TODO Auto-generated method stub

}

});

commitButton = (Button)findViewById(R.id.schduleManage\_commit);

commitButton.setOnClickListener(this);

}

@Override

public void onClick(View v) {

switch(v.getId()){

case R.id.datePickButton:

showDialog(DIALOG\_DATE); //if user click the calender, system show dailog

break;

case R.id.datePickButton2:

showDialog(DIALOG\_DATE2);//if user click the calender, system show dailog

break;

case R.id.schduleManage\_commit:

db = new DBmanager();

AlertDialog dialBox = createDialogBox();

dialBox.show();

String loc = menuActivity.locationCode;

//update location table

break;

}

}

private AlertDialog createDialogBox(){

AlertDialog myQuittingDialogBox =new AlertDialog.Builder(this)

//set message, title, and icon

.setTitle("Question")

.setMessage("湲곗〈�쓽 �벑濡앹씠 �궗�씪吏� �닔 �엳�뒿�땲�떎. 怨꾩냽 吏꾪뻾�븯�떆寃좎뒿�땲源�?")

.setIcon(R.drawable.ic\_launcher)

//set three option buttons

.setPositiveButton("Yes", new DialogInterface.OnClickListener() {

public void onClick(DialogInterface dialog, int whichButton) {

//whatever should be done when answering "YES" goes here

new openPartTimeTask().execute();

}

})//setPositiveButton

.setNegativeButton("NO", new DialogInterface.OnClickListener() {

public void onClick(DialogInterface dialog, int whichButton) {

//whatever should be done when answering "NO" goes here

}

})//setNegativeButton

.create();

return myQuittingDialogBox;

}// createDialogBox

// clas

@Override

protected Dialog onCreateDialog(int id){

if(id == DIALOG\_DATE)

return new DatePickerDialog(this, dateListener, 2015, 5, 1);

if(id == DIALOG\_DATE2)

return new DatePickerDialog(this, dateListener2, 2015, 5, 1);

return null;

}

private DatePickerDialog.OnDateSetListener dateListener =

new DatePickerDialog.OnDateSetListener() {

@Override

public void onDateSet(DatePicker view, int year, int monthOfYear,

int dayOfMonth) {

possibleDate = year+"-"+(monthOfYear+1)+"-"+dayOfMonth;

datePickButton.setText(year+"-"+(monthOfYear+1)+"-"+dayOfMonth);

}

};

private DatePickerDialog.OnDateSetListener dateListener2 =

new DatePickerDialog.OnDateSetListener() {

@Override

public void onDateSet(DatePicker view, int year, int monthOfYear,

int dayOfMonth) {

workDate = year+"-"+(monthOfYear+1)+"-"+dayOfMonth;

dateWorkButton.setText(year+"-"+(monthOfYear+1)+"-"+dayOfMonth);

}

};

//if manage click the setting date button, send query this information

private class openPartTimeTask extends AsyncTask<String, JSONArray, Void>{

private final ProgressDialog dialog = new ProgressDialog(Activity\_scheduleManage.this);

boolean check = false;

@Override

protected void onPreExecute() {

// TODO Auto-generated method stub

dialog.setMessage("�옞�떆留� 湲곕떎�젮 二쇱꽭�슂"); //show dialog

dialog.show();

super.onPreExecute();

}

@Override

protected Void doInBackground(String... params) {

//send query

String query = "update location\_info set state = '1', startDate = '" + possibleDate + "', endDate = '" + workDate + "', openNum = '" + openN + "', middleNum = '" + middleN + "', closeNum = '" + closeN + "' where locationCode = '" + menuActivity.locationCode + "';";

check = db.sendQuery(query);

return null;

}

@Override

protected void onPostExecute(Void result) {

if(check){ //state of complete

if(dialog.isShowing())

dialog.dismiss();

Toast.makeText(getApplicationContext(), "Complete.", Toast.LENGTH\_SHORT).show();

finish();

}else{ //state of default

if(dialog.isShowing())

dialog.dismiss();

Toast.makeText(getApplicationContext(), "Fail.", Toast.LENGTH\_SHORT).show();

}

super.onPostExecute(result);

}

}

}

**8. ApplyPerson.java**

**package** com.example.term;

//ApplyPerson object to use sorting and etc...

**public** **class** ApplyPerson {

**private** String ID;

**private** String name;

**private** String applyTime;

//Default constructor (Set each component)

**public** ApplyPerson(String newID, String newname, String newApplyTime)

{

ID = newID;

name = newname;

applyTime = newApplyTime;

}

//Return each component

**public** String getID()

{

**return** ID;

}

**public** String getName()

{

**return** name;

}

**public** String getApplyTime()

{

**return** applyTime;

}

//set each component

**public** **void** setID(String newID)

{

ID = newID;

}

**public** **void** setName(String newname)

{

name = newname;

}

**public** **void** setApplyTime(String newApplyTime)

{

applyTime = newApplyTime;

}

}

**9. CustomAdapter.java**

package com.example.term;

import java.util.ArrayList;

import android.content.Context;

import android.view.LayoutInflater;

import android.view.View;

import android.view.View.OnClickListener;

import android.view.ViewGroup;

import android.widget.BaseAdapter;

import android.widget.Button;

import android.widget.TextView;

//if manager click the newPeopleManage button, it supports this class

public class CustomAdapter extends BaseAdapter{

private ArrayList<String> name\_List;

private Context context = null;;

private OnClickListener myOnClickListener = null;

//Default constructor

public CustomAdapter()

{

name\_List = new ArrayList<String>();

}

//Default constructor

public CustomAdapter(OnClickListener onClickListener)

{

name\_List = new ArrayList<String>();

myOnClickListener = onClickListener;

}

//return componet

@Override

public int getCount() {

return name\_List.size();

}

@Override

public Object getItem(int position) {

return name\_List.get(position);

}

@Override

public long getItemId(int position) {

return position;

}

//make custom view [ Person image , list text, two button(Okay, Cancel) ]

@Override

public View getView(int position, View convertView, ViewGroup parent) {

final int pos = position;

context = parent.getContext();

TextView text = null;

Button acceptBtn = null;

Button rejectBtn = null;

CustomHolder holder = null;

String[] result = name\_List.get(position).split(" ");

String ID = result[0];

String name = result[1];

String openCode = result[2];

if ( convertView == null ) {

LayoutInflater inflater = (LayoutInflater) context.getSystemService(Context.LAYOUT\_INFLATER\_SERVICE);

convertView = inflater.inflate(R.layout.customadapter\_item, parent, false);

text = (TextView) convertView.findViewById(R.id.nameText);

acceptBtn = (Button) convertView.findViewById(R.id.acceptButton);

rejectBtn = (Button) convertView.findViewById(R.id.rejectButton);

holder = new CustomHolder();

holder.m\_TextView = text;

holder.ok\_Btn = acceptBtn;

holder.cancle\_Btn = rejectBtn;

if(openCode.equals("1"))

holder.ok\_Btn.setEnabled(false);

convertView.setTag(holder);

holder.ok\_Btn.setTag(pos);

holder.cancle\_Btn.setTag(pos);

}

else

{

holder = (CustomHolder) convertView.getTag();

text = holder.m\_TextView;

acceptBtn = holder.ok\_Btn;

rejectBtn = holder.cancle\_Btn;

acceptBtn.setTag(pos);

rejectBtn.setTag(pos);

if(openCode.equals("1"))

acceptBtn.setEnabled(false);

}

text.setText("ID : " + ID + " Name : " + name);

acceptBtn.setOnClickListener(myOnClickListener);

rejectBtn.setOnClickListener(myOnClickListener);

return convertView;

}

//add person to list

public void add(String name)

{

name\_List.add(name);

}

//remove person in list

public void remove(int position)

{

name\_List.remove(position);

}

private class CustomHolder{

TextView m\_TextView;

Button ok\_Btn;

Button cancle\_Btn;

}

}

**10. Dbmanager.java**

package com.example.term;

import java.io.BufferedReader;

import java.io.InputStream;

import java.io.InputStreamReader;

import java.util.ArrayList;

import org.apache.http.HttpEntity;

import org.apache.http.HttpResponse;

import org.apache.http.NameValuePair;

import org.apache.http.client.HttpClient;

import org.apache.http.client.entity.UrlEncodedFormEntity;

import org.apache.http.client.methods.HttpPost;

import org.apache.http.impl.client.DefaultHttpClient;

import org.apache.http.message.BasicNameValuePair;

import org.json.JSONArray;

import org.json.JSONException;

import android.util.Log;

//this class use access database. so, each object can request or receive query to DB

public class DBmanager {

String result = "";

ArrayList<NameValuePair> nameValuePairs;

//Default constructor

public DBmanager(){

nameValuePairs = new ArrayList<NameValuePair>();

}

//Part of request query

public JSONArray requestQuery(String sql){

InputStream is = null;

nameValuePairs.add(new BasicNameValuePair("sql",sql));

try{

//request to our database in apmsetup. using httpClient and httpPost

HttpClient httpclient = new DefaultHttpClient();

HttpPost httppost = new HttpPost("http://172.16.23.205:8080/getter\_mp.php");

UrlEncodedFormEntity entityRequest = new UrlEncodedFormEntity(nameValuePairs,"utf-8");

httppost.setEntity(entityRequest);

HttpResponse response = httpclient.execute(httppost);

HttpEntity entity = response.getEntity();

is = entity.getContent();

}catch(Exception e){

Log.e("error", "Error in http connection "+e.toString());

}

//convert response to string

try{

BufferedReader reader = new BufferedReader(new InputStreamReader(is,"utf-8"),8);

StringBuilder sb = new StringBuilder();

String line = null;

while ((line = reader.readLine()) != null) {

Log.e("error", "line: "+line);

sb.append(line + "\n");

}

is.close();

result=sb.toString();

}catch(Exception e){

Log.e("log\_tag", "Error converting result "+e.toString());

}

//parse json data

try{

if(result.equals("1\n")) Log.e("DBERROR", "connection error");

if(result.equals("2\n")) Log.e("DBERROR", "db selection error");

if(result.equals("3\n")) Log.e("DBERROR", "request error");

if(result.equals("4\n")) Log.e("DBERROR", "sql query error");

else {

JSONArray jArray = new JSONArray(result);

return jArray;

}

}catch(JSONException e){

Log.e("log\_tag", "Error parsing data "+e.toString());

}

return null;

}

public boolean sendQuery(String sql){

InputStream is = null;

nameValuePairs.add(new BasicNameValuePair("sql",sql));

//http post

try{

HttpClient httpclient = new DefaultHttpClient();

HttpPost httppost = new HttpPost("http://172.16.23.205:8080/setter\_mp.php");

httppost.setEntity(new UrlEncodedFormEntity(nameValuePairs,"utf-8"));

HttpResponse response = httpclient.execute(httppost);

HttpEntity entity = response.getEntity();

is = entity.getContent();

}catch(Exception e){

Log.e(null, "Error in http connection "+e.toString());

}

//convert response to string

try{

BufferedReader reader = new BufferedReader(new InputStreamReader(is,"utf-8"),8);

StringBuilder sb = new StringBuilder();

String line = null;

while ((line = reader.readLine()) != null) {

sb.append(line + "\n");

}

is.close();

result=sb.toString();

}catch(Exception e){

Log.e(null, "Error converting result "+e.toString());

}

if(result.equals("1\n"))

Log.e(null, "connection error");

if(result.equals("2\n"))

Log.e(null, "db selection error");

if(result.equals("3\n"))

Log.e(null, "request error");

if(result.equals("4\n"))

Log.e(null, "sql query error");

if(result.equals("5\n"))

{

Log.i("DB", "query success");

Log.e(null,"query success");

return false;

}

return true;

}

}

**11. MainActivity.java**

package com.example.term;

import org.json.JSONArray;

import org.json.JSONObject;

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.Menu;

import android.view.MenuItem;

import android.view.View;

import android.view.View.OnClickListener;

import android.widget.Button;

import android.widget.EditText;

import android.widget.TextView;

import android.widget.Toast;

//this class is login screen. if user want login or join our system, it supports this actions.

public class MainActivity extends Activity implements OnClickListener {

private TextView text;

private DBmanager db;

private EditText idEdit;

private EditText passEdit;

private String id;

private String pass;

private Button logBtn;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

//if system start, we show our splash screen

startActivity(new Intent("com.example.term.splash"));

getActionBar().hide();

db = new DBmanager();

idEdit = (EditText)findViewById(R.id.editID);

passEdit = (EditText)findViewById(R.id.editPasswrod);

logBtn = (Button)findViewById(R.id.buttonLogin);

text = (TextView)findViewById(R.id.newMember);

logBtn.setOnClickListener(this);

text.setOnClickListener(this);

}

public void menuClick(View v){

startActivity(new Intent("com.example.term.menuActivity"));

}

@Override

public boolean onCreateOptionsMenu(Menu menu) {

// Inflate the menu; this adds items to the action bar if it is present.

getMenuInflater().inflate(R.menu.main, menu);

return true;

}

@Override

public boolean onOptionsItemSelected(MenuItem item) {

// Handle action bar item clicks here. The action bar will

// automatically handle clicks on the Home/Up button, so long

// as you specify a parent activity in AndroidManifest.xml.

int id = item.getItemId();

if (id == R.id.action\_settings) {

return true;

}

return super.onOptionsItemSelected(item);

}

@Override

public void onClick(View v) {

if(v.getId() == text.getId()) //if user click join button, we show enrollment class

{

startActivity(new Intent(this,Activity\_enrollment.class));

}

else if(v.getId() == logBtn.getId() ) //if user click login button, we check ID & Password in DB

{

id = idEdit.getText().toString();

pass = passEdit.getText().toString();

if(id.length() == 0 || pass.length() == 0)

Toast.makeText(this, "ID & PASSWORD Reinput", Toast.LENGTH\_SHORT).show();

else

{

String query = "select \* from member\_info where ID = '" + id + "'and pass ='" + pass + "';";

new loginCheckTask().execute(query);

}

}

}

//it is used to check id and password correction

private class loginCheckTask extends AsyncTask<String, JSONArray, Void>{

private final ProgressDialog dialog = new ProgressDialog(MainActivity.this);

@Override

protected void onPreExecute() {

// TODO Auto-generated method stub

dialog.setMessage("濡쒓렇�씤 以묒엯�땲�떎."); //show dialog

dialog.show();

super.onPreExecute();

}

@Override

protected Void doInBackground(String... params) {

JSONArray idArray = db.requestQuery(params[0]);

publishProgress(idArray);

return null;

}

@Override

protected void onProgressUpdate(JSONArray... values) {

// TODO Auto-generated method stub

if(values[0].length() == 0)

Toast.makeText(getApplicationContext(), "Please check your ID & Password.", Toast.LENGTH\_SHORT).show();

else

{

Person myInfo;

try

{

JSONObject json\_data = values[0].getJSONObject(0);

String Name = json\_data.getString("name");

String ID= json\_data.getString("ID");

String password = json\_data.getString("pass");

String positionCode = json\_data.getString("positionCode");

String locationCode = json\_data.getString("locationCode");

String openCode = json\_data.getString("openCode");

myInfo = new Person(ID,password,Name,locationCode,positionCode, openCode);

Bundle myInfoBundle = new Bundle();

myInfoBundle.putSerializable("myInfo", myInfo);

Intent myIntent = new Intent("com.example.term.menuActivity");

myIntent.putExtras(myInfoBundle);

startActivity(myIntent);

Toast.makeText(getApplicationContext(), Name + "�떂 �솚�쁺�빀�땲�떎.", Toast.LENGTH\_SHORT).show();

finish();

}catch(Exception e)

{

Log.e("error",e.getMessage());

}

}

}

@Override

protected void onPostExecute(Void result) {

if(dialog.isShowing())

dialog.dismiss();

super.onPostExecute(result);

}

}

}

**12. menuActivity.java**

package com.example.term;

import org.json.JSONArray;

import org.json.JSONObject;

import android.app.ActionBar;

import android.app.Activity;

import android.content.Intent;

import android.os.AsyncTask;

import android.os.Bundle;

import android.os.Handler;

import android.util.Log;

import android.view.Menu;

import android.view.MenuItem;

import android.view.View;

import android.view.View.OnClickListener;

import android.widget.Button;

import android.widget.Toast;

//This class show menu button. if user click any button, we act predictable action

public class menuActivity extends Activity implements OnClickListener{

private Person myInfo;

private Button mySchedule, applyScedule, bulletineBoard, enrollmentScedule, fixSchedule;

private Button peopleManageBtn;

private Button backBtn;

private Intent myIntent;

private DBmanager db;

public static String openCode;

public static String locationCode;

public static String positionCode;

public static String name;

public static String ID;

public static String state;

private boolean bExit = false;

//initialize and store user information to use it other class

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.menu);

ActionBar a = getActionBar(); //action bar hide

a.hide();

db = new DBmanager();

//get Person information

myIntent = getIntent();

Bundle myBundle = myIntent.getExtras();

myInfo = (Person)myBundle.getSerializable("myInfo");

openCode = myInfo.getOpenCode();

locationCode = myInfo.getLocationCode();

positionCode = myInfo.getCode();

name = myInfo.getName();

ID = myInfo.getId();

new getStateTask().execute(); //get user information

//set id to use in java

applyScedule = (Button)findViewById(R.id.buttonApplyScedule);

mySchedule = (Button)findViewById(R.id.buttonMySchedule);

bulletineBoard = (Button)findViewById(R.id.buttonBulletinboard);

enrollmentScedule = (Button)findViewById(R.id.buttonEnrollmentSchedule);

peopleManageBtn = (Button)findViewById(R.id.buttonPeopleManage);

fixSchedule = (Button)findViewById(R.id.menu\_fixSchedule);

backBtn = (Button)findViewById(R.id.menu\_backBtn);

//set click listener

mySchedule.setOnClickListener(this);

applyScedule.setOnClickListener(this);

bulletineBoard.setOnClickListener(this);

enrollmentScedule.setOnClickListener(this);

peopleManageBtn.setOnClickListener(this);

fixSchedule.setOnClickListener(this);

backBtn.setOnClickListener(this);

}

@Override

public void onClick(View v) {

if(v.getId() == mySchedule.getId()){

if(openCode.equals("1")) //if user is accepted by manager, user can access

startActivity(new Intent(this, MySchedule.class));

else

Toast.makeText(this, "沅뚰븳�씠 �뾾�뒿�땲�떎 留ㅻ땲���뿉寃� 臾몄쓽�븯�꽭�슂.", Toast.LENGTH\_SHORT).show();

}

if(v.getId() == applyScedule.getId()){ //if user is manager then act, but any other doesn't show this class

if(positionCode.equals("1"))

startActivity(new Intent(this, Activity\_applyMySchedule.class));

else if(positionCode.equals("0") && openCode.equals("0"))

Toast.makeText(this, "沅뚰븳�씠 �뾾�뒿�땲�떎 留ㅻ땲���뿉寃� 臾몄쓽�븯�꽭�슂.", Toast.LENGTH\_SHORT).show();

else if(openCode.equals("1")&& state.equals("0") && positionCode.equals("0") )

Toast.makeText(this, "�떊泥� 湲곌컙�씠 �븘�떃�땲�떎.", Toast.LENGTH\_SHORT).show();

else if(openCode.equals("1")&& state.equals("1") && positionCode.equals("0"))

startActivity(new Intent(this, Activity\_applyMySchedule.class));

}

if(v.getId() == bulletineBoard.getId()){ //if user is accepted by manager, user can access

if(openCode.equals("1"))

startActivity(new Intent(this, Activity\_bulletin.class));

else

Toast.makeText(this, "沅뚰븳�씠 �뾾�뒿�땲�떎 留ㅻ땲���뿉寃� 臾몄쓽�븯�꽭�슂.", Toast.LENGTH\_SHORT).show();

}

if(v.getId() == enrollmentScedule.getId()){

if(positionCode.equalsIgnoreCase("1")){ //if user is manager then act, but any other doesn't show this class

startActivity(new Intent(this,Activity\_scheduleManage.class));

}

else

Toast.makeText(this, "留ㅻ땲��留� �궗�슜 媛��뒫�빀�땲�떎.", Toast.LENGTH\_SHORT).show();

}

if(v.getId() == peopleManageBtn.getId()){

if(positionCode.equalsIgnoreCase("1")) //if user is manager then act, but any other doesn't show this class

startActivity(new Intent(this, Activity\_newPeopleManage.class));

else

Toast.makeText(this, "留ㅻ땲��留� �궗�슜 媛��뒫�빀�땲�떎.", Toast.LENGTH\_SHORT).show();

}

if(v.getId() == fixSchedule.getId()){

if(positionCode.equalsIgnoreCase("1")) //if user is manager then act, but any other doesn't show this class

startActivity(new Intent(this, Activity\_fixSchedule.class));

else

Toast.makeText(this, "留ㅻ땲��留� �궗�슜 媛��뒫�빀�땲�떎", Toast.LENGTH\_SHORT).show();

}

if(v.getId() == backBtn.getId()) //if user click back button, user also must click back button.

{

if (bExit)

finish();

else {

Toast.makeText(this, "�븳踰� �뜑 �늻瑜댁떆硫� 醫낅즺�맗�땲�떎.",

Toast.LENGTH\_SHORT).show();

bExit = true;

new Handler().postDelayed(new Runnable() {

@Override

public void run() {

bExit = false;

}

}, 3 \* 1000);

}

}

}

@Override

public boolean onCreateOptionsMenu(Menu menu) {

// Inflate the menu; this adds items to the action bar if it is present.

getMenuInflater().inflate(R.menu.main, menu);

return true;

}

@Override

public boolean onOptionsItemSelected(MenuItem item) {

// Handle action bar item clicks here. The action bar will

// automatically handle clicks on the Home/Up button, so long

// as you specify a parent activity in AndroidManifest.xml.

int id = item.getItemId();

if (id == R.id.action\_settings) {

return true;

}

return super.onOptionsItemSelected(item);

}

@Override

public void onBackPressed() { //if user click back button, user also must click back button.

if (bExit)

finish();

else {

Toast.makeText(this, "�븳踰� �뜑 �늻瑜댁떆硫� 醫낅즺�맗�땲�떎.",

Toast.LENGTH\_SHORT).show();

bExit = true;

new Handler().postDelayed(new Runnable() {

@Override

public void run() {

bExit = false;

}

}, 3 \* 1000);

}

}

//get user information and using any class

private class getStateTask extends AsyncTask<String, JSONArray, Void>{

@Override

protected Void doInBackground(String... params) {

//send query and get information

JSONArray idArray = db.requestQuery("select state from location\_info where locationCode = '" + locationCode + "';");

publishProgress(idArray);

return null;

}

@Override

protected void onProgressUpdate(JSONArray... values) {

// TODO Auto-generated method stub

try

{

JSONObject json\_data = values[0].getJSONObject(0);

state = json\_data.getString("state");

//get data through JSONObject

}catch(Exception e)

{

Log.e("error",e.getMessage());

}

}

}

}

**13. MySchedule.java**

package com.example.term;

import java.util.ArrayList;

import org.json.JSONArray;

import org.json.JSONException;

import org.json.JSONObject;

import android.app.Activity;

import android.app.DatePickerDialog;

import android.app.Dialog;

import android.app.ProgressDialog;

import android.os.AsyncTask;

import android.os.Bundle;

import android.util.Log;

import android.view.View;

import android.view.View.OnClickListener;

import android.widget.ArrayAdapter;

import android.widget.Button;

import android.widget.DatePicker;

import android.widget.ListView;

import android.widget.TextView;

//This class show P-Timer schedule

public class MySchedule extends Activity implements OnClickListener {

private static final int DIALOG\_DATE = 0;

private DBmanager db;

private Button dateSelectBtn;

private Button finishbtn;

private TextView titleTextView;

private ListView lv1;

private ArrayList<String> list ;

private ArrayAdapter<String> adapter;

private String startYear, startMonth, startDays;

private String endYear, endMonth, endDays;

private String pickDate;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_myschedule);

getActionBar().hide();

db = new DBmanager();

//set id and use in java

dateSelectBtn = (Button)findViewById(R.id.datePick);

titleTextView = (TextView)findViewById(R.id.dateText);

finishbtn = (Button)findViewById(R.id.rotation\_okBtn);

lv1 = (ListView)findViewById(R.id.workList1);

list = new ArrayList<String>();

//set adapter

adapter = new ArrayAdapter<String>(this, android.R.layout.simple\_list\_item\_1,list);

lv1.setAdapter(adapter);

//set click listener

dateSelectBtn.setOnClickListener(this);

finishbtn.setOnClickListener(this);

new getDateTask().execute(menuActivity.locationCode);

}

@Override

public void onClick(View v) {

// TODO Auto-generated method stub

if(v.getId() == dateSelectBtn.getId()){

showDialog(DIALOG\_DATE); //show calender

}

if(v.getId() == finishbtn.getId())

finish();

}

protected Dialog onCreateDialog(int id) {

DatePickerDialog \_date = null;

switch (id) { //show date of limited by manager

case DIALOG\_DATE:

\_date = new DatePickerDialog(this, dateListener,

Integer.parseInt(startYear), Integer.parseInt(startMonth)-1, Integer.parseInt(startDays)){

@Override

public void onDateChanged(DatePicker view, int year, int monthOfYear, int dayOfMonth){

if (year > Integer.parseInt(endYear))

view.updateDate(Integer.parseInt(endYear), Integer.parseInt(endMonth) - 1, Integer.parseInt(endDays));

if (monthOfYear > Integer.parseInt(endMonth) - 1 && year == Integer.parseInt(endYear))

view.updateDate(Integer.parseInt(endYear), Integer.parseInt(endMonth) - 1, Integer.parseInt(endDays));

if (dayOfMonth > Integer.parseInt(endDays) && year == Integer.parseInt(endYear) && monthOfYear == Integer.parseInt(endMonth) - 1)

view.updateDate(Integer.parseInt(endYear), Integer.parseInt(endMonth) - 1, Integer.parseInt(endDays));

//these below lines of code used for setting the maximum as well as minimum dates on Date Picker Dialog..

if (year < Integer.parseInt(startYear))

view.updateDate(Integer.parseInt(startYear), Integer.parseInt(startMonth) - 1, Integer.parseInt(startDays));

if (monthOfYear < Integer.parseInt(startMonth) - 1 && year == Integer.parseInt(startYear) )

view.updateDate(Integer.parseInt(startYear), Integer.parseInt(startMonth) - 1, Integer.parseInt(startDays));

if (dayOfMonth < Integer.parseInt(startDays) && year == Integer.parseInt(startYear) && monthOfYear == Integer.parseInt(startMonth) - 1)

view.updateDate(Integer.parseInt(startYear), Integer.parseInt(startMonth) - 1, Integer.parseInt(startDays));

}

};

}

return \_date;

}

//if user click date, set date

private DatePickerDialog.OnDateSetListener dateListener = new DatePickerDialog.OnDateSetListener() {

@Override

public void onDateSet(DatePicker view, int year, int monthOfYear,

int dayOfMonth) {

pickDate = year+"-"+(monthOfYear+1)+"-"+dayOfMonth;

dateSelectBtn.setText(year+"-"+(monthOfYear+1)+"-"+dayOfMonth);

new getTask().execute(menuActivity.locationCode);

}

};

//get task in DB

private class getTask extends AsyncTask<String, JSONArray, Void>{

private final ProgressDialog dialog = new ProgressDialog(MySchedule.this);

@Override

protected void onPreExecute() {

// TODO Auto-generated method stub

dialog.setMessage("�옞�떆留� 湲곕떎�젮 二쇱꽭�슂.");

dialog.show(); //show dialog

super.onPreExecute();

}

@Override

protected Void doInBackground(String... params) {

String query = "select \* from fixed\_info where locationCode = '" + params[0] +

"' and workDate = '" + pickDate +"' and name = '" + menuActivity.name + "';";

//send query and get P-Timer work information

JSONArray workInfo = db.requestQuery(query);

try {

list.clear();

for(int i = 0; i < workInfo.length(); i++){

JSONObject workOBJ = workInfo.getJSONObject(i);

String caseOfWork = workOBJ.getString("fixedTime");

//set list of each state

if(caseOfWork.equals("1")){

list.add("Open : " + workOBJ.getString("name"));

}if(caseOfWork.equals("12")){

list.add("Open : " +workOBJ.getString("name"));

list.add("Middle : " +workOBJ.getString("name"));

}if(caseOfWork.equals("13")){

list.add("Open : " +workOBJ.getString("name"));

list.add("Close : " +workOBJ.getString("name"));

}if(caseOfWork.equals("123")){

Log.e("mySchedule","+99");

list.add("Open : " +workOBJ.getString("name"));

list.add("Middle : " +workOBJ.getString("name"));

list.add("Close : " +workOBJ.getString("name"));

Log.e("mySchedule","+1010");

}if(caseOfWork.equals("2")){

list.add("Middle : " +workOBJ.getString("name"));

}if(caseOfWork.equals("23")){

list.add("Middle : " +workOBJ.getString("name"));

list.add("Close : " +workOBJ.getString("name"));

}if(caseOfWork.equals("3")){

list.add("Close : " +workOBJ.getString("name"));

}

}

} catch (JSONException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

publishProgress();

return null;

}

@Override

protected void onProgressUpdate(JSONArray... values) {

super.onProgressUpdate(values);

adapter.notifyDataSetChanged();

}

@Override

protected void onPostExecute(Void result) {

if(dialog.isShowing())

dialog.dismiss(); //dismiss dialog

super.onPostExecute(result);

}

}

//get date of limited by manager

private class getDateTask extends AsyncTask<String, JSONArray, Void>{

private final ProgressDialog dialog = new ProgressDialog(MySchedule.this);

@Override

protected void onPreExecute() {

// TODO Auto-generated method stub

dialog.setMessage("�옞�떆留� 湲곕떎�젮 二쇱꽭�슂.");

dialog.show(); //show dialog

super.onPreExecute();

}

@Override

protected Void doInBackground(String... params) {

String query = "select \* from location\_info where locationCode = '" + params[0] + "';";

JSONArray dateInfo = db.requestQuery(query);

publishProgress(dateInfo);

return null;

}

@Override

protected void onProgressUpdate(JSONArray... values) {

super.onProgressUpdate(values);

try

{

JSONObject json\_data = values[0].getJSONObject(0);

//get object and get data through this

String startDate = json\_data.getString("startDate");

String endDate= json\_data.getString("endDate");

String temp[] = startDate.split("-");

String temp2[] = endDate.split("-");

startYear = temp[0];

startMonth = temp[1];

startDays = temp[2];

endYear = temp2[0];

endMonth = temp2[1];

endDays = temp2[2];

}catch(Exception e){

Log.e("error",e.getMessage());

}

}

@Override

protected void onPostExecute(Void result) {

if(dialog.isShowing())

dialog.dismiss(); //dismiss dialog

super.onPostExecute(result);

}

}

}

**15. Person.java**

**package** com.example.term;

**import** java.io.Serializable;

**public** **class** Person **implements** Serializable{

**private** String id;

**private** String password;

**private** String name;

**private** String locationCode;

**private** String positionCode;

**private** String openCode;

//default constructor

**public** Person()

{

}

//default constructor and set component

**public** Person(String id, String password, String name, String locationCode, String code, String openCode)

{

**this**.id = id;

**this**.password = password;

**this**.name = name;

**this**.locationCode = locationCode;

**this**.positionCode = code;

**this**.openCode = openCode;

}

//set component

**public** **void** setId(String newId)

{

id = newId;

}

**public** **void** setPassword(String newPassword)

{

password = newPassword;

}

**public** **void** setName(String newName)

{

name = newName;

}

**public** **void** setLocationCode(String code)

{

locationCode = code;

}

**public** **void** setPositionCode(String newCode)

{

positionCode = newCode;

}

**public** **void** setOpenCode(String openCode)

{

**this**.openCode = openCode;

}

//return component

**public** String getOpenCode()

{

**return** openCode;

}

**public** String getId()

{

**return** id;

}

**public** String getPassword()

{

**return** password;

}

**public** String getName()

{

**return** name;

}

**public** String getLocationCode()

{

**return** locationCode;

}

**public** String getCode()

{

**return** positionCode;

}

}

**16. splash.java**

package com.example.term;

import android.app.Activity;

import android.os.Bundle;

import android.os.Handler;

//it show our logo. it is spalsh screen

public class splash extends Activity{

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.splash);

getActionBar().hide(); //actionbar hide

Handler hd = new Handler();

hd.postDelayed(new Runnable(){

@Override

public void run(){

finish(); //it is finished by system

}

}, 3500);

}

}

**- XML CODE**

**1. activity\_applymyschedule.xml**

<?xml version=*"1.0"* encoding=*"utf-8"*?>

<LinearLayout xmlns:android=*"http://schemas.android.com/apk/res/android"*

android:layout\_width=*"fill\_parent"*

android:layout\_height=*"fill\_parent"*

android:background=*"@drawable/backimg"*

android:orientation=*"vertical"* >

<LinearLayout

android:id=*"@+id/dueday"*

android:layout\_width=*"match\_parent"*

android:layout\_height=*"wrap\_content"*

android:layout\_margin=*"5dip"*

android:layout\_marginTop=*"10dp"*

android:background=*"#80F9F9F9"*

android:orientation=*"vertical"* >

<LinearLayout

android:layout\_width=*"match\_parent"*

android:layout\_height=*"wrap\_content"*

android:layout\_margin=*"5dip"*

android:background=*"#80F9F9F9"*

android:orientation=*"vertical"* >

<TextView

android:id=*"@+id/dateText"*

android:layout\_width=*"wrap\_content"*

android:layout\_height=*"wrap\_content"*

android:text=*"Set date"*

android:textColor=*"#ffffff"*

android:textSize=*"25sp"*

android:textStyle=*"bold"* />

<TextView

android:id=*"@+id/applyduration"*

android:layout\_width=*"wrap\_content"*

android:layout\_height=*"wrap\_content"*

android:layout\_marginLeft=*"80dp"*

android:text=*"TextView"*

android:textColor=*"#ffffff"*

android:textSize=*"15sp"* />

<LinearLayout

android:layout\_width=*"wrap\_content"*

android:layout\_height=*"wrap\_content"*

android:orientation=*"horizontal"* >

<TextView

android:id=*"@+id/applyScheduleTitle"*

android:layout\_width=*"150dp"*

android:layout\_height=*"50dp"*

android:gravity=*"center"*

android:text=*"Work Date : "*

android:textAlignment=*"center"*

android:textSize=*"18dp"*

android:textStyle=*"bold"* />

<Button

android:id=*"@+id/selectDataBtn"*

android:layout\_width=*"150dp"*

android:layout\_height=*"50dp"*

android:layout\_gravity=*"right"*

android:layout\_marginLeft=*"80dp"*

android:background=*"#00999999"*

android:drawableLeft=*"@drawable/calendaricon2"*

android:text=*" "* />

</LinearLayout>

</LinearLayout>

</LinearLayout>

<LinearLayout

android:layout\_width=*"match\_parent"*

android:layout\_height=*"400dp"*

android:layout\_margin=*"5dp"*

android:layout\_marginTop=*"10dp"*

android:background=*"#80F9F9F9"*

android:orientation=*"vertical"* >

<LinearLayout

android:id=*"@+id/linearLayout"*

android:layout\_width=*"match\_parent"*

android:layout\_height=*"400dp"*

android:layout\_margin=*"5dp"*

android:background=*"#80F9F9F9"*

android:orientation=*"vertical"* >

<LinearLayout

android:layout\_width=*"match\_parent"*

android:layout\_height=*"133dp"*

android:orientation=*"horizontal"* >

<TextView

android:id=*"@+id/openText"*

android:layout\_width=*"150dp"*

android:layout\_height=*"wrap\_content"*

android:layout\_gravity=*"center"*

android:layout\_marginLeft=*"5dp"*

android:shadowColor=*"#7000"*

android:shadowDx=*"12"*

android:shadowDy=*"12"*

android:shadowRadius=*"8"*

android:text=*"Open time"*

android:textSize=*"18dp"*

android:textStyle=*"bold"* />

<CheckBox

android:id=*"@+id/openCheckBox"*

android:layout\_width=*"200dp"*

android:layout\_height=*"match\_parent"*

android:onClick=*"onCheckBoxClicked"*

android:paddingTop=*"60dp"* />

</LinearLayout>

<LinearLayout

android:layout\_width=*"match\_parent"*

android:layout\_height=*"133dp"*

android:orientation=*"horizontal"* >

<TextView

android:id=*"@+id/middleText"*

android:layout\_width=*"150dp"*

android:layout\_height=*"wrap\_content"*

android:layout\_gravity=*"center"*

android:layout\_marginLeft=*"5dp"*

android:shadowColor=*"#7000"*

android:shadowDx=*"12"*

android:shadowDy=*"12"*

android:shadowRadius=*"8"*

android:text=*"Middle time"*

android:textSize=*"18dp"*

android:textStyle=*"bold"* />

<CheckBox

android:id=*"@+id/middleCheckBox"*

android:layout\_width=*"wrap\_content"*

android:layout\_height=*"wrap\_content"*

android:onClick=*"onCheckBoxClicked"*

android:paddingTop=*"60dp"* />

</LinearLayout>

<LinearLayout

android:layout\_width=*"match\_parent"*

android:layout\_height=*"133dp"*

android:orientation=*"horizontal"* >

<TextView

android:id=*"@+id/closeText"*

android:layout\_width=*"150dp"*

android:layout\_height=*"wrap\_content"*

android:layout\_gravity=*"center"*

android:layout\_marginLeft=*"5dp"*

android:shadowColor=*"#7000"*

android:shadowDx=*"12"*

android:shadowDy=*"12"*

android:shadowRadius=*"8"*

android:text=*"Close time"*

android:textSize=*"18dp"*

android:textStyle=*"bold"* />

<CheckBox

android:id=*"@+id/closeCheckBox"*

android:layout\_width=*"wrap\_content"*

android:layout\_height=*"wrap\_content"*

android:onClick=*"onCheckBoxClicked"*

android:paddingTop=*"60dp"* />

</LinearLayout>

</LinearLayout>

</LinearLayout>

<LinearLayout

android:layout\_width=*"wrap\_content"*

android:layout\_height=*"wrap\_content"*

android:layout\_gravity=*"center"* >

<Button

android:id=*"@+id/saveBtn"*

android:layout\_width=*"100dp"*

android:layout\_height=*"60dp"*

android:layout\_below=*"@+id/linearLayout"*

android:layout\_centerHorizontal=*"true"*

android:layout\_marginTop=*"20dp"*

android:background=*"@drawable/save"*

android:gravity=*"center"*

android:textSize=*"20dp"* />

</LinearLayout>

</LinearLayout>

**2. activity\_bulletin.xml**

<?xml version=*"1.0"* encoding=*"utf-8"*?>

<LinearLayout xmlns:android=*"http://schemas.android.com/apk/res/android"*

android:layout\_width=*"match\_parent"*

android:layout\_height=*"match\_parent"*

android:orientation=*"vertical"* >

<WebView

android:id=*"@+id/webView"*

android:layout\_width=*"match\_parent"*

android:layout\_height=*"390dp"*

android:layout\_weight=*"1.39"*

>

</WebView>

<Button

android:id=*"@+id/homeButton"*

android:layout\_width=*"match\_parent"*

android:layout\_height=*"wrap\_content"*

android:background=*"@drawable/homebtn"* />

</LinearLayout>

**3. activity\_enrollment.xml**

<?xml version=*"1.0"* encoding=*"utf-8"*?>

<LinearLayout xmlns:android=*"http://schemas.android.com/apk/res/android"*

android:layout\_width=*"match\_parent"*

android:layout\_height=*"match\_parent"*

android:background=*"@drawable/backimg"*

android:orientation=*"vertical"*

android:weightSum=*"1"* >

<TextView

android:id=*"@+id/textView1"*

android:layout\_width=*"wrap\_content"*

android:layout\_height=*"wrap\_content"*

android:layout\_gravity=*"center"*

android:text=*"Welcome to join Part - Timer - Scheduler"*

android:textColor=*"#ffffff"*

android:textSize=*"15dp"*

android:textStyle=*"italic"*

android:typeface=*"serif"* />

<TableLayout

android:layout\_width=*"match\_parent"*

android:layout\_height=*"300dp"*

android:layout\_weight=*"1"*

android:orientation=*"vertical"*

android:stretchColumns=*"\*"* >

<View

android:layout\_width=*"match\_parent"*

android:layout\_height=*"1dp"* />

<TableRow

android:layout\_width=*"match\_parent"*

android:layout\_height=*"wrap\_content"*

android:paddingBottom=*"34dp"* >

<TextView

android:layout\_width=*"wrap\_content"*

android:layout\_height=*"wrap\_content"*

android:layout\_margin=*"1dp"*

android:autoText=*"false"*

android:gravity=*"center"*

android:padding=*"3dp"*

android:text=*"ID"*

android:textAlignment=*"center"*

android:textColor=*"#ffffff"*

android:textSize=*"20dp"*

android:textStyle=*"bold"* />

<EditText

android:id=*"@+id/idEditText"*

android:layout\_width=*"wrap\_content"*

android:layout\_height=*"wrap\_content"*

android:layout\_margin=*"1dp"*

android:hint=*"Enter ID"*

android:padding=*"3dp"*

android:textColor=*"#ffffff"*

android:textColorHint=*"#ffffff"* >

</EditText>

<Button

android:id=*"@+id/idCheckButton"*

android:layout\_width=*"match\_parent"*

android:layout\_height=*"wrap\_content"*

android:layout\_margin=*"1dp"*

android:gravity=*"center"*

android:padding=*"3dp"*

android:text=*"중복확인"*

android:textColor=*"#ffffff"* />

</TableRow>

<TableRow

android:layout\_width=*"match\_parent"*

android:layout\_height=*"wrap\_content"*

android:paddingBottom=*"34dp"* >

<TextView

android:layout\_width=*"wrap\_content"*

android:layout\_height=*"wrap\_content"*

android:layout\_margin=*"1dp"*

android:autoText=*"false"*

android:gravity=*"center"*

android:padding=*"3dp"*

android:text=*"PW"*

android:textAlignment=*"center"*

android:textColor=*"#ffffff"*

android:textSize=*"20dp"*

android:textStyle=*"bold"* />

<EditText

android:id=*"@+id/passEditText"*

android:layout\_width=*"wrap\_content"*

android:layout\_height=*"wrap\_content"*

android:layout\_margin=*"1dp"*

android:layout\_span=*"2"*

android:hint=*"Enter Password"*

android:inputType=*"textPassword"*

android:padding=*"3dp"*

android:textColor=*"#ffffff"*

android:textColorHint=*"#ffffff"* >

</EditText>

</TableRow>

<TableRow

android:layout\_width=*"match\_parent"*

android:layout\_height=*"wrap\_content"*

android:paddingBottom=*"34dp"* >

<TextView

android:layout\_width=*"wrap\_content"*

android:layout\_height=*"wrap\_content"*

android:layout\_margin=*"1dp"*

android:autoText=*"false"*

android:gravity=*"center"*

android:padding=*"3dp"*

android:text=*" "*

android:textAlignment=*"center"*

android:textColor=*"#ffffff"*

android:textSize=*"20dp"*

android:textStyle=*"bold"* />

<EditText

android:id=*"@+id/passEditText2"*

android:layout\_width=*"wrap\_content"*

android:layout\_height=*"wrap\_content"*

android:layout\_margin=*"1dp"*

android:layout\_span=*"2"*

android:hint=*"Enter Password Again"*

android:inputType=*"textPassword"*

android:padding=*"3dp"*

android:textColor=*"#ffffff"*

android:textColorHint=*"#ffffff"* >

</EditText>

</TableRow>

<TableRow

android:layout\_width=*"match\_parent"*

android:layout\_height=*"wrap\_content"*

android:paddingBottom=*"34dp"* >

<TextView

android:layout\_width=*"wrap\_content"*

android:layout\_height=*"wrap\_content"*

android:layout\_margin=*"1dp"*

android:autoText=*"false"*

android:gravity=*"center"*

android:padding=*"3dp"*

android:text=*"Name"*

android:textAlignment=*"center"*

android:textColor=*"#ffffff"*

android:textSize=*"20dp"*

android:textStyle=*"bold"* />

<EditText

android:id=*"@+id/nameEditText"*

android:layout\_width=*"wrap\_content"*

android:layout\_height=*"wrap\_content"*

android:layout\_margin=*"1dp"*

android:layout\_span=*"2"*

android:hint=*"Enter Your Name"*

android:padding=*"3dp"*

android:textColor=*"#ffffff"*

android:textColorHint=*"#ffffff"* >

</EditText>

</TableRow>

<TableRow

android:layout\_width=*"match\_parent"*

android:layout\_height=*"wrap\_content"*

android:paddingBottom=*"34dp"* >

<TextView

android:layout\_width=*"wrap\_content"*

android:layout\_height=*"wrap\_content"*

android:layout\_margin=*"1dp"*

android:autoText=*"false"*

android:gravity=*"center"*

android:padding=*"3dp"*

android:text=*"Type"*

android:textAlignment=*"center"*

android:textColor=*"#ffffff"*

android:textSize=*"20dp"*

android:textStyle=*"bold"* />

<RadioGroup

android:id=*"@+id/radioGroup"*

android:layout\_width=*"wrap\_content"*

android:layout\_height=*"wrap\_content"*

android:layout\_gravity=*"center"*

android:layout\_span=*"2"*

android:orientation=*"horizontal"* >

<RadioButton

android:id=*"@+id/radioBtnTimer"*

android:layout\_width=*"wrap\_content"*

android:layout\_height=*"wrap\_content"*

android:text=*"알바생"*

android:textColor=*"#ffffff"* />

<RadioButton

android:id=*"@+id/radioBtnManager"*

android:layout\_width=*"wrap\_content"*

android:layout\_height=*"wrap\_content"*

android:text=*"매니저"*

android:textColor=*"#ffffff"* />

</RadioGroup>

</TableRow>

<TableRow

android:layout\_width=*"match\_parent"*

android:layout\_height=*"wrap\_content"*

android:paddingBottom=*"34dp"* >

<TextView

android:layout\_width=*"wrap\_content"*

android:layout\_height=*"wrap\_content"*

android:layout\_margin=*"1dp"*

android:autoText=*"false"*

android:gravity=*"center"*

android:padding=*"3dp"*

android:text=*"매장Code"*

android:textAlignment=*"center"*

android:textColor=*"#ffffff"*

android:textSize=*"20dp"*

android:textStyle=*"bold"* />

<EditText

android:id=*"@+id/locationEditText"*

android:layout\_width=*"wrap\_content"*

android:layout\_height=*"wrap\_content"*

android:layout\_margin=*"1dp"*

android:hint=*"Enter Your Shop Code"*

android:padding=*"3dp"*

android:textColor=*"#ffffff"*

android:textColorHint=*"#ffffff"* />

<Button

android:id=*"@+id/shopCheckBtn"*

android:layout\_width=*"match\_parent"*

android:layout\_height=*"wrap\_content"*

android:layout\_margin=*"1dp"*

android:gravity=*"center"*

android:padding=*"3dp"*

android:text=*"확인"*

android:textColor=*"#ffffff"* />

</TableRow>

</TableLayout>

<Button

android:id=*"@+id/OKButton9"*

android:layout\_width=*"150dp"*

android:layout\_height=*"80dp"*

android:layout\_gravity=*"center\_horizontal"*

android:background=*"@drawable/commitbtn"* />

</LinearLayout>

**4. activity\_fixschedule.xml**

<?xml version=*"1.0"* encoding=*"utf-8"*?>

<LinearLayout xmlns:android=*"http://schemas.android.com/apk/res/android"*

android:layout\_width=*"match\_parent"*

android:layout\_height=*"match\_parent"*

android:background=*"@drawable/backimg"*

android:orientation=*"vertical"* >

<LinearLayout

android:layout\_width=*"match\_parent"*

android:layout\_height=*"110dp"*

android:layout\_margin=*"5dip"*

android:background=*"#80F9F9F9"*

android:orientation=*"vertical"* >

<LinearLayout

android:layout\_width=*"match\_parent"*

android:layout\_height=*"wrap\_content"*

android:orientation=*"horizontal"* >

<TextView

android:layout\_width=*"match\_parent"*

android:layout\_height=*"wrap\_content"*

android:text=*"Apply List"*

android:textSize=*"25sp"*

android:textStyle=*"bold"* >

</TextView>

</LinearLayout>

<LinearLayout

android:layout\_width=*"match\_parent"*

android:layout\_height=*"wrap\_content"*

android:orientation=*"horizontal"* >

<Button

android:id=*"@+id/fixScedule\_showCatalogBtn"*

android:layout\_width=*"35dp"*

android:layout\_height=*"35dp"*

android:layout\_margin=*"5dip"*

android:background=*"@drawable/calendaricon2"* />

<TextView

android:id=*"@+id/fixSchedule\_dateText"*

android:layout\_width=*"40dp"*

android:layout\_height=*"40dp"*

android:layout\_gravity=*"center"*

android:layout\_marginLeft=*"50dp"*

android:gravity=*"center"*

android:text=*""*

android:textSize=*"15dp"* >

</TextView>

</LinearLayout>

<LinearLayout

android:layout\_width=*"wrap\_content"*

android:layout\_height=*"wrap\_content"*

android:orientation=*"horizontal"* >

<Button

android:id=*"@+id/fixScedule\_showfixmemberList"*

android:layout\_width=*"35dp"*

android:layout\_height=*"35dp"*

android:layout\_marginLeft=*"5dip"*

android:background=*"@drawable/rotabtn"* />

</LinearLayout>

</LinearLayout>

<LinearLayout

android:layout\_width=*"match\_parent"*

android:layout\_height=*"110dp"*

android:layout\_margin=*"5dip"*

android:background=*"#80F9F9F9"*

android:orientation=*"vertical"* >

<TextView

android:id=*"@+id/fix\_textView1"*

android:layout\_width=*"wrap\_content"*

android:layout\_height=*"wrap\_content"*

android:text=*"Open Time"*

android:textColor=*"#ffffff"*

android:textSize=*"15dp"*

android:textStyle=*"bold"* />

<ListView

android:id=*"@+id/fixScedule\_open"*

android:layout\_width=*"match\_parent"*

android:layout\_height=*"110dp"*

android:layout\_margin=*"5dip"*

android:background=*"#80F9F9F9"* >

</ListView>

</LinearLayout>

<LinearLayout

android:layout\_width=*"match\_parent"*

android:layout\_height=*"110dp"*

android:layout\_margin=*"5dip"*

android:background=*"#80F9F9F9"*

android:orientation=*"vertical"* >

<TextView

android:id=*"@+id/fix\_textView2"*

android:layout\_width=*"wrap\_content"*

android:layout\_height=*"wrap\_content"*

android:layout\_margin=*"5dip"*

android:text=*"Middle Time"*

android:textColor=*"#ffffff"*

android:textSize=*"15dp"*

android:textStyle=*"bold"* />

<ListView

android:id=*"@+id/fixScedule\_middle"*

android:layout\_width=*"match\_parent"*

android:layout\_height=*"100dp"*

android:layout\_margin=*"5dip"*

android:background=*"#80F9F9F9"* >

</ListView>

</LinearLayout>

<LinearLayout

android:layout\_width=*"match\_parent"*

android:layout\_height=*"110dp"*

android:layout\_margin=*"5dip"*

android:background=*"#80F9F9F9"*

android:orientation=*"vertical"* >

<TextView

android:id=*"@+id/fix\_textView3"*

android:layout\_width=*"wrap\_content"*

android:layout\_height=*"wrap\_content"*

android:layout\_margin=*"5dip"*

android:text=*"Close Time"*

android:textColor=*"#ffffff"*

android:textSize=*"15dp"*

android:textStyle=*"bold"* />

<ListView

android:id=*"@+id/fixScedule\_close"*

android:layout\_width=*"match\_parent"*

android:layout\_height=*"100dp"*

android:layout\_margin=*"5dip"*

android:background=*"#80F9F9F9"* >

</ListView>

</LinearLayout>

</LinearLayout>

**5. activity\_main.xml**

<RelativeLayout xmlns:android=*"http://schemas.android.com/apk/res/android"*

xmlns:tools=*"http://schemas.android.com/tools"*

android:layout\_width=*"match\_parent"*

android:layout\_height=*"match\_parent"*

android:background=*"@drawable/backimg"*

android:paddingBottom=*"@dimen/activity\_vertical\_margin"*

android:paddingLeft=*"@dimen/activity\_horizontal\_margin"*

android:paddingRight=*"@dimen/activity\_horizontal\_margin"*

android:paddingTop=*"@dimen/activity\_vertical\_margin"*

tools:context=*"com.example.term.MainActivity"* >

<ImageView

android:id=*"@+id/imageView1"*

android:layout\_width=*"match\_parent"*

android:layout\_height=*"178dp"*

android:layout\_weight=*"0.83"*

android:src=*"@drawable/title"* />

<EditText

android:id=*"@+id/editID"*

android:layout\_width=*"match\_parent"*

android:layout\_height=*"wrap\_content"*

android:layout\_alignBottom=*"@+id/imageView1"*

android:layout\_alignRight=*"@+id/imageView1"*

android:ems=*"10"*

android:gravity=*"left"*

android:hint=*"ID"*

android:textColor=*"#ffffff"*

android:textColorHint=*"#ffffff"* />

<EditText

android:id=*"@+id/editPasswrod"*

android:layout\_width=*"match\_parent"*

android:layout\_height=*"wrap\_content"*

android:layout\_alignLeft=*"@+id/editID"*

android:layout\_below=*"@+id/editID"*

android:ems=*"10"*

android:gravity=*"left"*

android:hint=*"Password "*

android:password=*"true"*

android:singleLine=*"true"*

android:textColor=*"#ffffff"*

android:textColorHint=*"#ffffff"* />

<TextView

android:id=*"@+id/newMember"*

android:layout\_width=*"wrap\_content"*

android:layout\_height=*"wrap\_content"*

android:layout\_alignParentBottom=*"true"*

android:layout\_centerHorizontal=*"true"*

android:layout\_gravity=*"center"*

android:layout\_marginBottom=*"37dp"*

android:onClick=*""*

android:text=*"Sign up"*

android:textColor=*"#ffffff"*

android:textSize=*"15dp"*

android:textStyle=*"bold"* />

<Button

android:id=*"@+id/buttonLogin"*

android:layout\_width=*"match\_parent"*

android:layout\_height=*"wrap\_content"*

android:layout\_below=*"@+id/editPasswrod"*

android:layout\_marginTop=*"16dp"*

android:background=*"#44ffffff"*

android:gravity=*"center"*

android:onClick=*"menuClick"*

android:text=*"Login"* />

</RelativeLayout>

**6. activity\_myschedule.xml**

<?xml version=*"1.0"* encoding=*"utf-8"*?>

<LinearLayout xmlns:android=*"http://schemas.android.com/apk/res/android"*

android:layout\_width=*"match\_parent"*

android:layout\_height=*"match\_parent"*

android:background=*"@drawable/backimg"*

android:orientation=*"vertical"* >

<LinearLayout

android:layout\_width=*"match\_parent"*

android:layout\_height=*"100dp"*

android:layout\_margin=*"5dip"*

android:background=*"#80F9F9F9"*

android:gravity=*"center"*

android:orientation=*"horizontal"* >

<LinearLayout

android:layout\_width=*"match\_parent"*

android:layout\_height=*"match\_parent"*

android:layout\_margin=*"5dip"*

android:background=*"#80F9F9F9"*

android:gravity=*"center"* >

<Button

android:id=*"@+id/datePick"*

android:layout\_width=*"70dp"*

android:layout\_height=*"70dp"*

android:layout\_margin=*"15dip"*

android:background=*"@drawable/calendaricon"*

android:textColor=*"#ffffff"*

android:textSize=*"25dp"* />

<TextView

android:id=*"@+id/dateText"*

android:layout\_width=*"wrap\_content"*

android:layout\_height=*"wrap\_content"*

android:layout\_margin=*"20dip"*

android:text=*"Date"*

android:textSize=*"35dp"*

android:textStyle=*"bold"* />

</LinearLayout>

</LinearLayout>

<LinearLayout

android:layout\_width=*"match\_parent"*

android:layout\_height=*"350dp"*

android:layout\_margin=*"5dip"*

android:background=*"#80F9F9F9"*

android:orientation=*"vertical"* >

<TextView

android:id=*"@+id/mySchedule\_text"*

android:layout\_width=*"wrap\_content"*

android:layout\_height=*"wrap\_content"*

android:layout\_margin=*"5dip"*

android:text=*"My Schedule"*

android:textColor=*"#ffffff"*

android:textSize=*"15dp"*

android:textStyle=*"bold"* />

<ListView

android:id=*"@+id/workList1"*

android:layout\_width=*"wrap\_content"*

android:layout\_height=*"wrap\_content"*

android:layout\_margin=*"5dip"*

android:background=*"#80F9F9F9"* >

</ListView>

</LinearLayout>

<LinearLayout

android:layout\_width=*"match\_parent"*

android:layout\_height=*"wrap\_content"*

android:gravity=*"center"* >

<Button

android:id=*"@+id/rotation\_okBtn"*

android:layout\_width=*"100dp"*

android:layout\_height=*"60dp"*

android:layout\_gravity=*"center"*

android:background=*"@drawable/tempbtn"* />

</LinearLayout>

</LinearLayout>

**7. activity\_newpeoplemanage.xml**

<?xml version=*"1.0"* encoding=*"utf-8"*?>

<LinearLayout xmlns:android=*"http://schemas.android.com/apk/res/android"*

android:layout\_width=*"match\_parent"*

android:layout\_height=*"match\_parent"*

android:background=*"@drawable/backimg"*

android:orientation=*"vertical"* >

<ListView

android:id=*"@+id/peopleListView"*

android:layout\_width=*"match\_parent"*

android:layout\_height=*"wrap\_content"* >

</ListView>

</LinearLayout>

**8. activity\_rotationapplicant.xml**

<?xml version=*"1.0"* encoding=*"utf-8"*?>

<LinearLayout xmlns:android=*"http://schemas.android.com/apk/res/android"*

android:layout\_width=*"match\_parent"*

android:layout\_height=*"match\_parent"*

android:background=*"@drawable/backimg"*

android:orientation=*"vertical"* >

<LinearLayout

android:layout\_width=*"match\_parent"*

android:layout\_height=*"wrap\_content"*

android:gravity=*"right"*

android:layout\_margin=*"5dip"*

android:background=*"#80F9F9F9"*

android:orientation=*"horizontal"* >

<Button

android:id=*"@+id/rotation\_roBtn"*

android:layout\_width=*"50dp"*

android:layout\_height=*"50dp"*

android:background=*"@drawable/rotabtn"* />

</LinearLayout>

<LinearLayout

android:layout\_width=*"match\_parent"*

android:layout\_height=*"wrap\_content"*

android:layout\_margin=*"5dip"*

android:background=*"#80F9F9F9"*

android:orientation=*"vertical"* >

<TextView

android:id=*"@+id/fix\_textView1"*

android:layout\_width=*"wrap\_content"*

android:layout\_height=*"wrap\_content"*

android:layout\_margin=*"5dip"*

android:text=*"Open Time"*

android:textColor=*"#ffffff"*

android:textSize=*"15dp"*

android:textStyle=*"bold"* />

<ListView

android:id=*"@+id/rotation\_opneList"*

android:layout\_width=*"match\_parent"*

android:layout\_height=*"120dp"*

android:layout\_margin=*"5dip"*

android:layout\_weight=*"1"*

android:background=*"#80F9F9F9"* >

</ListView>

</LinearLayout>

<LinearLayout

android:layout\_width=*"match\_parent"*

android:layout\_height=*"wrap\_content"*

android:layout\_margin=*"5dip"*

android:background=*"#80F9F9F9"*

android:orientation=*"vertical"* >

<TextView

android:id=*"@+id/fix\_textView2"*

android:layout\_width=*"wrap\_content"*

android:layout\_height=*"wrap\_content"*

android:layout\_margin=*"5dip"*

android:text=*"Middle Time"*

android:textColor=*"#ffffff"*

android:textSize=*"15dp"*

android:textStyle=*"bold"* />

<ListView

android:id=*"@+id/rotation\_middleList"*

android:layout\_width=*"match\_parent"*

android:layout\_height=*"120dp"*

android:layout\_margin=*"5dip"*

android:layout\_weight=*"1"*

android:background=*"#80F9F9F9"* >

</ListView>

</LinearLayout>

<LinearLayout

android:layout\_width=*"match\_parent"*

android:layout\_height=*"wrap\_content"*

android:layout\_margin=*"5dip"*

android:background=*"#80F9F9F9"*

android:orientation=*"vertical"* >

<TextView

android:id=*"@+id/fix\_textView3"*

android:layout\_width=*"wrap\_content"*

android:layout\_height=*"wrap\_content"*

android:layout\_margin=*"5dip"*

android:text=*"Close Time"*

android:textColor=*"#ffffff"*

android:textSize=*"15dp"*

android:textStyle=*"bold"* />

<ListView

android:id=*"@+id/rotation\_closeList"*

android:layout\_width=*"match\_parent"*

android:layout\_height=*"120dp"*

android:layout\_margin=*"5dip"*

android:layout\_weight=*"1"*

android:background=*"#80F9F9F9"* >

</ListView>

</LinearLayout>

<LinearLayout

android:layout\_width=*"match\_parent"*

android:layout\_height=*"wrap\_content"*

android:gravity=*"center"* >

<Button

android:id=*"@+id/rotation\_okBtn"*

android:layout\_width=*"100dp"*

android:layout\_height=*"60dp"*

android:layout\_gravity=*"center"*

android:background=*"@drawable/commitbtn"* />

</LinearLayout>

</LinearLayout>

**9. activity\_schedulemanage.xml**

<?xml version=*"1.0"* encoding=*"utf-8"*?>

<LinearLayout xmlns:android=*"http://schemas.android.com/apk/res/android"*

android:layout\_width=*"fill\_parent"*

android:layout\_height=*"fill\_parent"*

android:background=*"@drawable/backimg"*

android:orientation=*"vertical"* >

<LinearLayout

android:id=*"@+id/dueday"*

android:layout\_width=*"match\_parent"*

android:layout\_height=*"150dp"*

android:layout\_margin=*"5dp"*

android:background=*"#80F9F9F9"*

android:orientation=*"vertical"* >

<LinearLayout

android:layout\_width=*"match\_parent"*

android:layout\_height=*"wrap\_content"*

android:layout\_margin=*"5dp"*

android:background=*"#80F9F9F9"*

android:orientation=*"vertical"* >

<TextView

android:id=*"@id/dateText"*

android:layout\_width=*"match\_parent"*

android:layout\_height=*"30dp"*

android:text=*"Set date"*

android:textColor=*"#ffffff"*

android:textSize=*"25sp"*

android:textStyle=*"bold"* />

<LinearLayout

android:layout\_width=*"match\_parent"*

android:layout\_height=*"wrap\_content"*

android:layout\_margin=*"5dp"*

android:gravity=*"center"*

android:orientation=*"horizontal"* >

<TextView

android:layout\_width=*"100dp"*

android:layout\_height=*"20dp"*

android:text=*"Start Date : "*

android:textSize=*"18dp"*

android:textStyle=*"bold"* />

<Button

android:id=*"@+id/datePickButton"*

android:layout\_width=*"200dp"*

android:layout\_height=*"50dp"*

android:background=*"#00999999"*

android:drawableLeft=*"@drawable/calendaricon2"*

android:text=*" "* />

</LinearLayout>

<LinearLayout

android:layout\_width=*"match\_parent"*

android:layout\_height=*"wrap\_content"*

android:layout\_margin=*"5dp"*

android:gravity=*"center"*

android:orientation=*"horizontal"* >

<TextView

android:layout\_width=*"100dp"*

android:layout\_height=*"20dp"*

android:text=*"End Date : "*

android:textSize=*"18dp"*

android:textStyle=*"bold"* />

<Button

android:id=*"@+id/datePickButton2"*

android:layout\_width=*"200dp"*

android:layout\_height=*"50dp"*

android:background=*"#00999999"*

android:drawableLeft=*"@drawable/calendaricon2"*

android:text=*" "* />

</LinearLayout>

</LinearLayout>

</LinearLayout>

<LinearLayout

android:layout\_width=*"match\_parent"*

android:layout\_height=*"300dp"*

android:layout\_margin=*"5dp"*

android:background=*"#80F9F9F9"*

android:orientation=*"vertical"* >

<LinearLayout

android:layout\_width=*"match\_parent"*

android:layout\_height=*"90dp"*

android:layout\_margin=*"5dp"*

android:background=*"#80F9F9F9"*

android:orientation=*"vertical"* >

<TextView

android:id=*"@+id/openText"*

android:layout\_width=*"150dp"*

android:layout\_height=*"wrap\_content"*

android:layout\_gravity=*"left"*

android:text=*"Number Of Open"*

android:textColor=*"#ffffff"*

android:textSize=*"18dp"*

android:textStyle=*"bold"* />

<Spinner

android:id=*"@+id/openNumber"*

android:layout\_width=*"100dp"*

android:layout\_height=*"90dp"*

android:layout\_gravity=*"center"*

android:text=*"Open number of people"* />

</LinearLayout>

<LinearLayout

android:layout\_width=*"match\_parent"*

android:layout\_height=*"90dp"*

android:layout\_margin=*"5dp"*

android:background=*"#80F9F9F9"*

android:orientation=*"vertical"* >

<TextView

android:id=*"@+id/middleText"*

android:layout\_width=*"150dp"*

android:layout\_height=*"wrap\_content"*

android:layout\_gravity=*"left"*

android:text=*"Number Of Open"*

android:textColor=*"#ffffff"*

android:textSize=*"18dp"*

android:textStyle=*"bold"* />

<Spinner

android:id=*"@+id/middleNumber"*

android:layout\_width=*"100dp"*

android:layout\_height=*"90dp"*

android:layout\_gravity=*"center"*

android:text=*"Open number of people"* />

</LinearLayout>

<LinearLayout

android:layout\_width=*"match\_parent"*

android:layout\_height=*"90dp"*

android:layout\_margin=*"5dp"*

android:background=*"#80F9F9F9"*

android:orientation=*"vertical"* >

<TextView

android:id=*"@+id/closeText"*

android:layout\_width=*"150dp"*

android:layout\_height=*"wrap\_content"*

android:layout\_gravity=*"left"*

android:text=*"Number Of Open"*

android:textColor=*"#ffffff"*

android:textSize=*"18dp"*

android:textStyle=*"bold"* />

<Spinner

android:id=*"@+id/closeNumber"*

android:layout\_width=*"100dp"*

android:layout\_height=*"90dp"*

android:layout\_gravity=*"center"*

android:text=*"Open number of people"* />

</LinearLayout>

</LinearLayout>

<LinearLayout

android:layout\_width=*"wrap\_content"*

android:layout\_height=*"wrap\_content"*

android:layout\_gravity=*"center"* >

<Button

android:id=*"@+id/schduleManage\_commit"*

android:layout\_width=*"100dp"*

android:layout\_height=*"60dp"*

android:layout\_gravity=*"center"*

android:background=*"@drawable/commitbtn"*

android:textSize=*"20dp"* />

</LinearLayout>

</LinearLayout>

**10. customadapter\_item.xml**

<?xml version=*"1.0"* encoding=*"utf-8"*?>

<LinearLayout xmlns:android=*"http://schemas.android.com/apk/res/android"*

android:layout\_width=*"match\_parent"*

android:layout\_height=*"40dp"*

android:orientation=*"horizontal"*

android:layout\_marginTop=*"2dp"*

android:background=*"#80000000"*

android:weightSum=*"1"*>

<ImageView

android:layout\_width=*"40dp"*

android:layout\_height=*"40dp"*

android:id=*"@+id/mem\_picture"*

android:layout\_weight=*"0.35"*

android:src=*"@drawable/people"*

android:background=*"#80F9F9F9"*/>

<TextView

android:id=*"@+id/nameText"*

android:layout\_width=*"80dp"*

android:layout\_height=*"40dp"*

android:text=*"name"*

android:textAlignment=*"gravity"*

android:textColor=*"#ffffff"*

android:gravity=*"center"*

android:layout\_weight=*"0.62"*

android:autoText=*"false"* />

<LinearLayout

android:layout\_width=*"wrap\_content"*

android:layout\_height=*"40dp"*

android:orientation=*"horizontal"*

android:baselineAligned=*"false"*>

<Button

android:id=*"@+id/acceptButton"*

android:layout\_width=*"60dp"*

android:layout\_height=*"40dp"*

android:gravity=*"center"*

android:textSize=*"15dp"*

android:background=*"@drawable/ac1"* />

<Button

android:id=*"@+id/rejectButton"*

android:layout\_width=*"60dp"*

android:gravity=*"center"*

android:textSize=*"15dp"*

android:layout\_height=*"40dp"*

android:background=*"@drawable/de1"* />

</LinearLayout>

</LinearLayout>

**11. menu.xml**

<LinearLayout xmlns:android=*"http://schemas.android.com/apk/res/android"*

xmlns:tools=*"http://schemas.android.com/tools"*

android:layout\_width=*"match\_parent"*

android:layout\_height=*"match\_parent"*

android:background=*"@drawable/backimg"*

android:orientation=*"vertical"*

android:paddingBottom=*"@dimen/activity\_vertical\_margin"*

android:paddingLeft=*"@dimen/activity\_horizontal\_margin"*

android:paddingRight=*"@dimen/activity\_horizontal\_margin"*

android:paddingTop=*"@dimen/activity\_vertical\_margin"*

tools:context=*"com.example.term.MainActivity"* >

<LinearLayout

android:layout\_width=*"match\_parent"*

android:layout\_height=*"wrap\_content"*

android:orientation=*"horizontal"* >

<Button

android:id=*"@+id/buttonBulletinboard"*

android:layout\_width=*"95dp"*

android:layout\_height=*"122dp"*

android:layout\_margin=*"5dip"*

android:background=*"@drawable/btnback1"* />

<Button

android:id=*"@+id/buttonMySchedule"*

android:layout\_width=*"215dp"*

android:layout\_height=*"122dp"*

android:layout\_margin=*"5dip"*

android:background=*"@drawable/btnback4"* />

</LinearLayout>

<LinearLayout

android:layout\_width=*"match\_parent"*

android:layout\_height=*"wrap\_content"*

android:orientation=*"horizontal"* >

<Button

android:id=*"@+id/buttonEnrollmentSchedule"*

android:layout\_width=*"215dp"*

android:layout\_height=*"122dp"*

android:layout\_margin=*"5dip"*

android:background=*"@drawable/btnback6"* />

<Button

android:id=*"@+id/buttonPeopleManage"*

android:layout\_width=*"95dp"*

android:layout\_height=*"122dp"*

android:layout\_margin=*"5dip"*

android:background=*"@drawable/btnback3"* />

</LinearLayout>

<LinearLayout

android:layout\_width=*"match\_parent"*

android:layout\_height=*"wrap\_content"*

android:orientation=*"horizontal"* >

<Button

android:id=*"@+id/menu\_fixSchedule"*

android:layout\_width=*"95dp"*

android:layout\_height=*"122dp"*

android:layout\_margin=*"5dip"*

android:background=*"@drawable/btnback2"* />

<AnalogClock

android:id=*"@+id/analogClock1"*

android:layout\_width=*"match\_parent"*

android:layout\_height=*"match\_parent"* />

</LinearLayout>

<LinearLayout

android:layout\_width=*"match\_parent"*

android:layout\_height=*"wrap\_content"*

android:orientation=*"horizontal"* >

<Button

android:id=*"@+id/buttonApplyScedule"*

android:layout\_width=*"215dp"*

android:layout\_height=*"122dp"*

android:layout\_margin=*"5dip"*

android:background=*"@drawable/btnback5"* />

<Button

android:id=*"@+id/menu\_backBtn"*

android:layout\_width=*"95dp"*

android:layout\_height=*"122dp"*

android:layout\_margin=*"5dip"*

android:background=*"@drawable/btnback7"* />

</LinearLayout>

</LinearLayout>

**12. myschedulemain.xml**

<?xml version=*"1.0"* encoding=*"utf-8"*?>

<RelativeLayout xmlns:android=*"http://schemas.android.com/apk/res/android"*

android:layout\_width=*"match\_parent"*

android:layout\_height=*"match\_parent"* >

<RelativeLayout

android:id=*"@+id/layout1"*

android:layout\_width=*"wrap\_content"*

android:layout\_height=*"wrap\_content"*

android:orientation=*"vertical"* >

<TextView

android:id=*"@+id/dateText"*

android:layout\_width=*"wrap\_content"*

android:layout\_height=*"wrap\_content"*

android:layout\_alignParentLeft=*"true"*

android:layout\_alignParentRight=*"true"*

android:text=*"TextView"* />

<Button

android:id=*"@+id/datePick"*

android:layout\_width=*"wrap\_content"*

android:layout\_height=*"wrap\_content"*

android:layout\_alignParentLeft=*"true"*

android:layout\_alignParentRight=*"true"*

android:layout\_below=*"@+id/textView1"*

android:text=*"Button"* />

<ListView

android:id=*"@+id/workList1"*

android:layout\_width=*"wrap\_content"*

android:layout\_height=*"wrap\_content"*

android:layout\_alignParentLeft=*"true"*

android:layout\_alignParentRight=*"true"*

android:layout\_below=*"@+id/datePick"* >

</ListView>

</RelativeLayout>

</RelativeLayout>

**13. splash.xml**

<RelativeLayout xmlns:android=*"http://schemas.android.com/apk/res/android"*

xmlns:tools=*"http://schemas.android.com/tools"*

android:layout\_width=*"match\_parent"*

android:layout\_height=*"match\_parent"*

android:background=*"@drawable/backimg"*

android:paddingBottom=*"@dimen/activity\_vertical\_margin"*

android:paddingLeft=*"@dimen/activity\_horizontal\_margin"*

android:paddingRight=*"@dimen/activity\_horizontal\_margin"*

android:paddingTop=*"@dimen/activity\_vertical\_margin"*

tools:context=*"com.example.term.MainActivity"* >

<ImageView

android:id=*"@+id/logoview"*

android:layout\_width=*"wrap\_content"*

android:layout\_height=*"wrap\_content"*

android:layout\_alignParentLeft=*"true"*

android:layout\_alignParentTop=*"true"*

android:layout\_marginTop=*"79dp"*

android:src=*"@drawable/title"* />

</RelativeLayout>