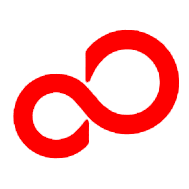
**DEPLOY GUIDE APPS**



**MOBILE WORKING REPORT**

**FUJITSU INDONESIA**

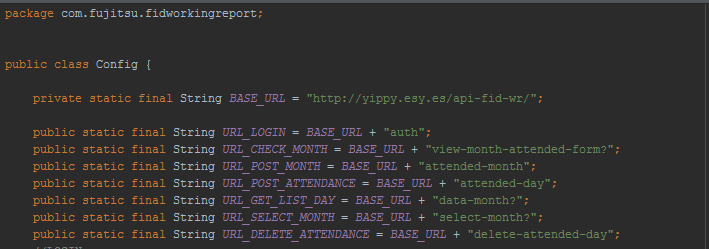
# Tools instalation

* Android Studio 2.1.2
* JDK 8
* SDK version 23
* Android Smartphone vers : 5.1 (lolipop) / android emulator

# Configure Apps in Android Studio

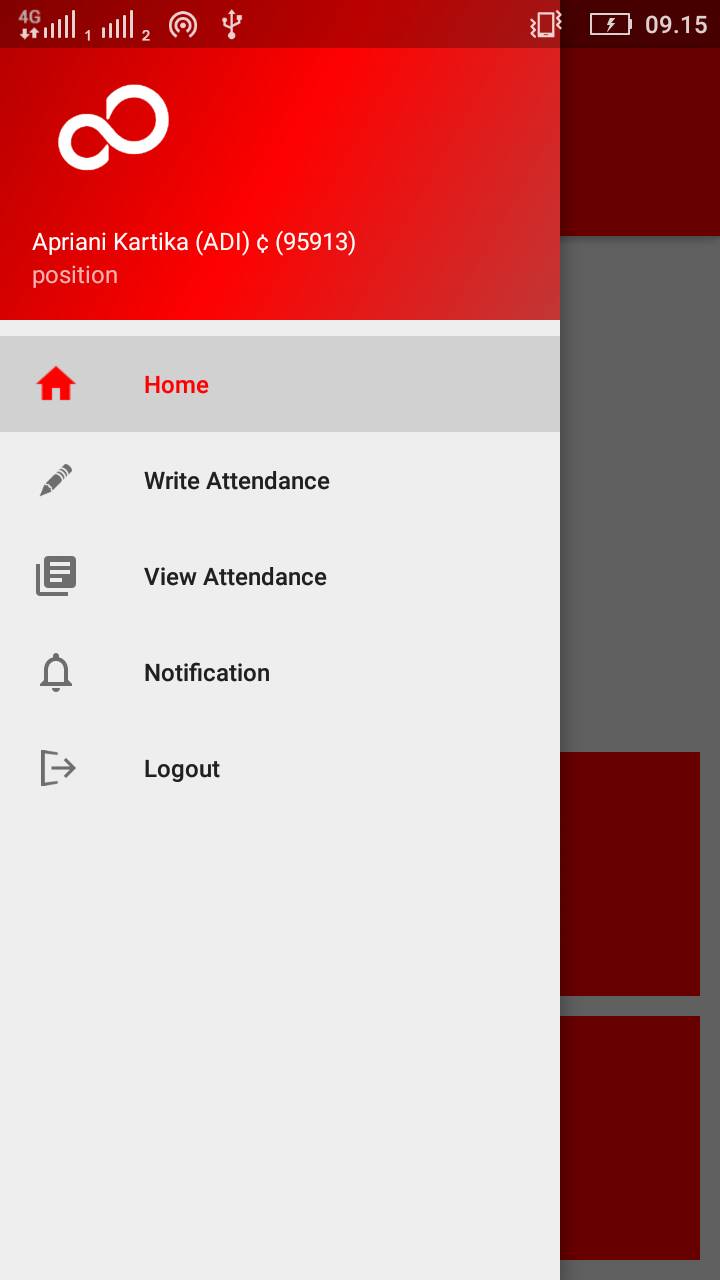
## Setup API link

File java : **Config.java**



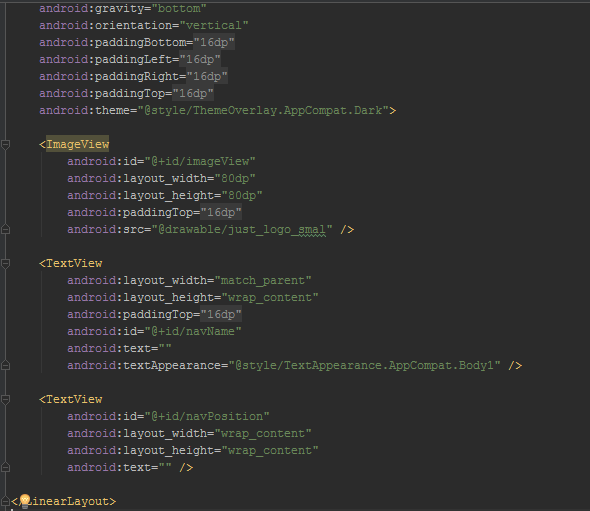
* BASE\_URL : Base url API for request to server. If server moved, BASE\_URL must be change
* URL\_LOGIN : Url API for login
* URL\_CHECK\_MONTH : URL for get data monthly information and applied for monthly information form
* URL\_POST\_MONTH : URL for post monthly information
* URL\_POST\_ATTENDANCE : URL for post daily attendance
* URL\_GET\_LIST\_DAY : URL for get list all of day in a month
* URL\_SELECT\_MONTH : URL for View Attendance it will get list all of month in a year
* URL\_DELETE\_ATTENDANCE : URL for delete daily attendance

## Drawer Navigation



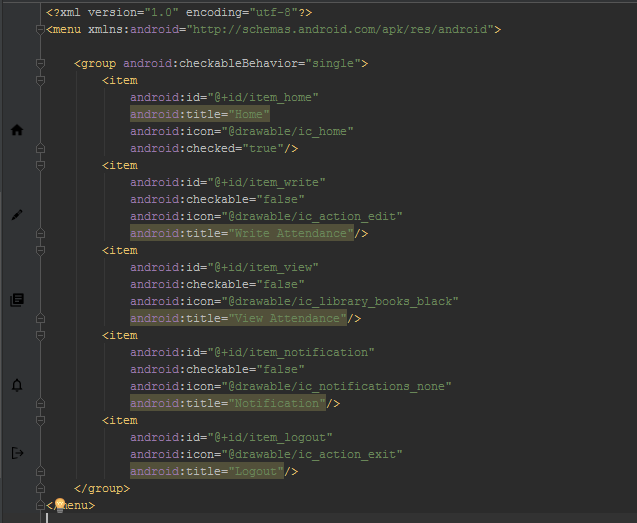
Drawer navigation divided into navigation header and navigation menu

### C:\Users\Ryan Baskara\AppData\Local\Microsoft\Windows\INetCache\Content.Word\drawerView.jpgNav Header

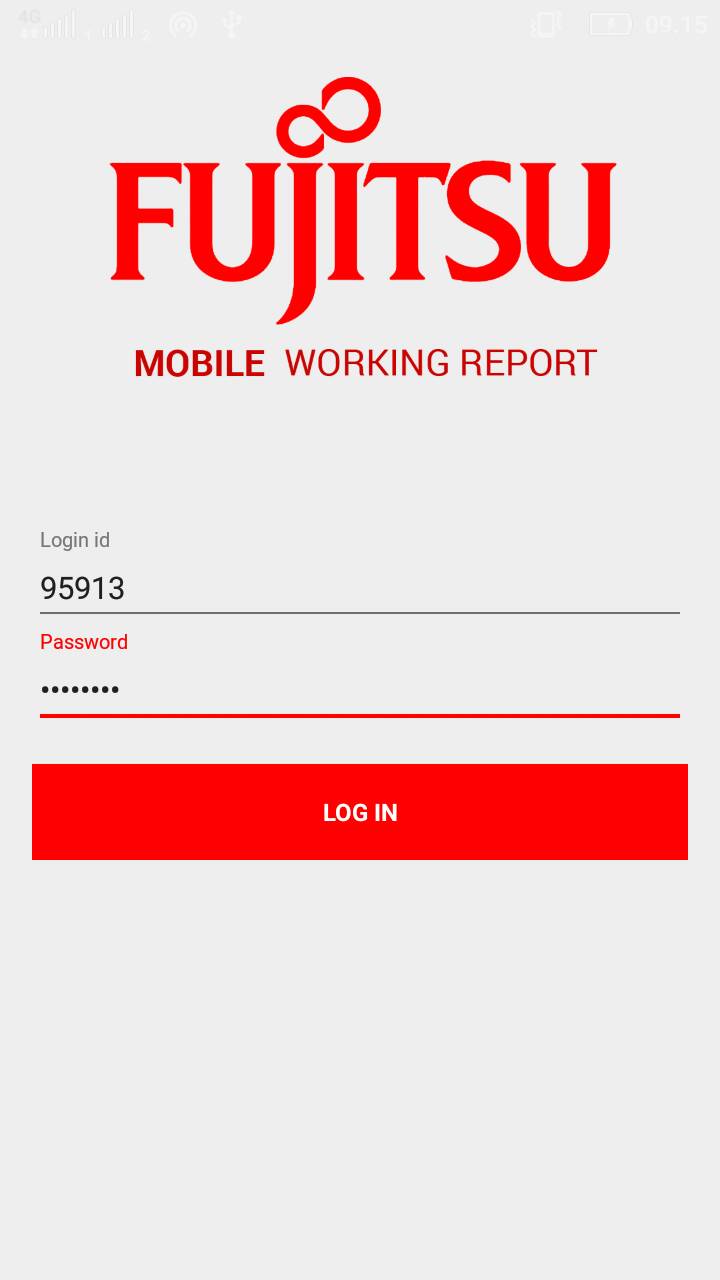
To modifie nav header view go to **res\layout\nav\_header\_main.xml**

* ImageView with id **imageView** usedto display image in nav header
* TextView with id navName used to display employee name in nav header
* TextView with id navPosition used to display position of employee in nav header

### C:\Users\Ryan Baskara\AppData\Local\Microsoft\Windows\INetCache\Content.Word\drawerView.jpgNavigation Menu

To modifie navigation menu go to **res\menu\activity\_main\_drawer.xml**

## Login page

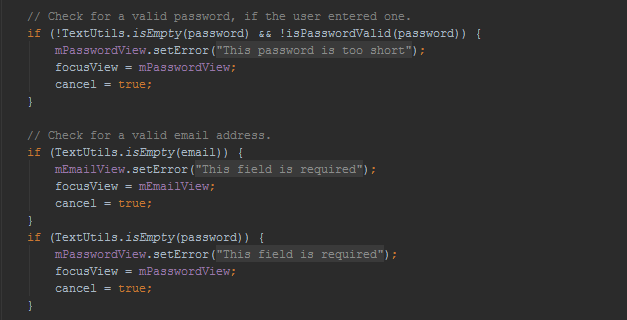


File java : LoginActivity.java

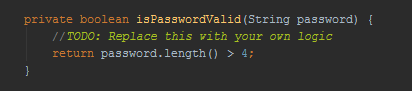
File XML : activity\_login.xml

### Check Validation

To check validation go to function attempLogin()

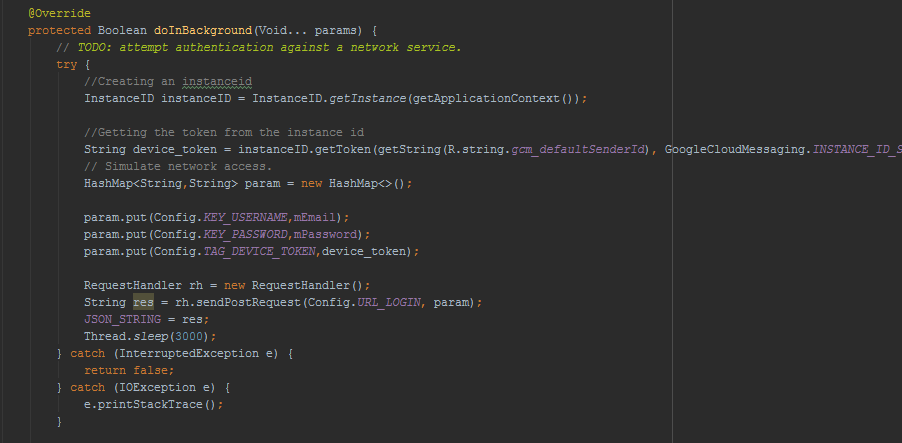


It make login\_id and pasword required



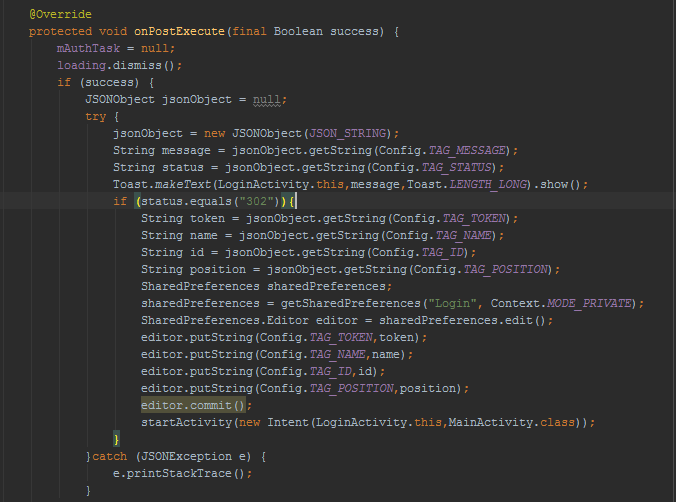
It make password length at least 4 characters

### Send request to server

Function name : doInBackground(Void... params)

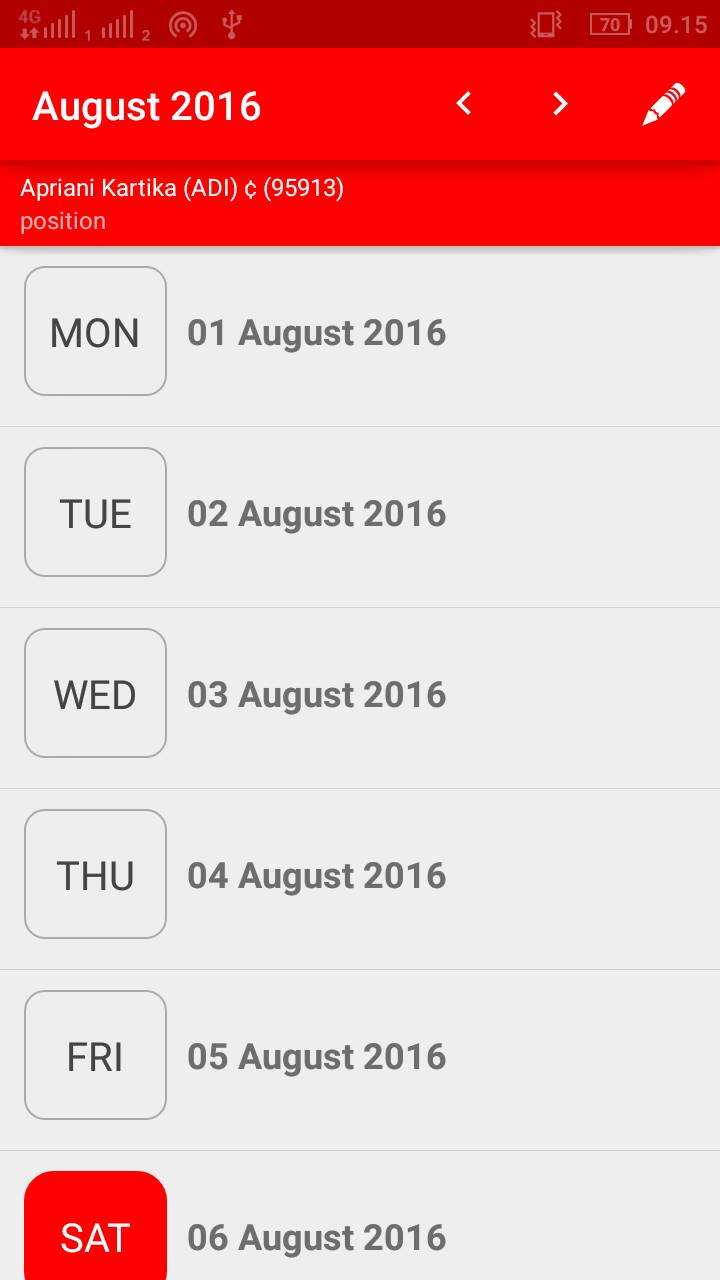
It will send value of login\_id, password, and device token. Device token is used to push notification. RequestHandler is the class for send request to server.

### Save data to local storage



The request will be return token, name, id, position, status. Status 302 means the user is found. SharePreference used to save data to local storage.

## Write Attendance

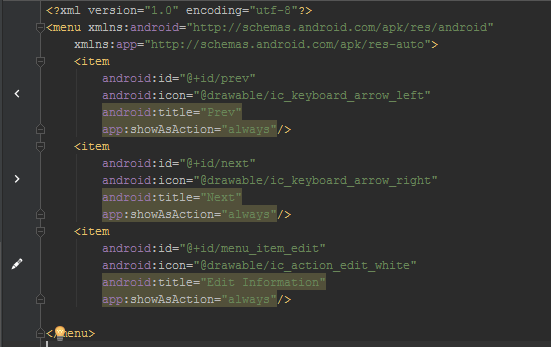


File Java : WriteAttendance.java

File xml : activity\_list.xml

### Toolbar

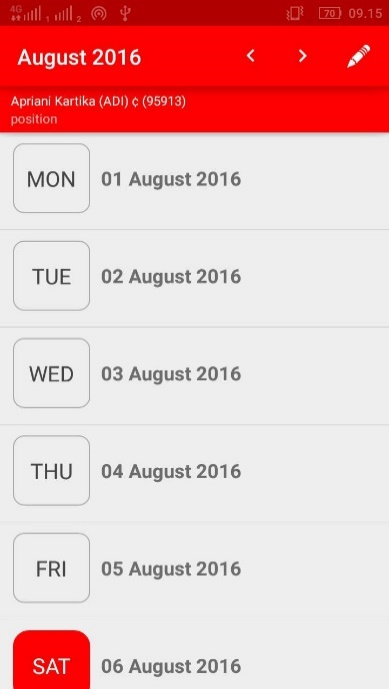
To set toolbar icon go to file in res\menu\list\_day.xml



To set action of icon in WriteAttendance.java in function onOptionsItemSelected(MenuItem item)

@Override  
public boolean onOptionsItemSelected(MenuItem item){  
  
 int temp\_month = Integer.*parseInt*(month);  
 int tmp\_year = Integer.*parseInt*(year);  
 switch(item.getItemId()){  
 case R.id.*prev*:  
 if (temp\_month==1){  
 temp\_month = 12;  
 tmp\_year = tmp\_year -1;  
 }else{  
 temp\_month = temp\_month - 1;  
 }  
  
 Intent intent1 = new Intent(this, WriteAttendance.class);  
 intent1.putExtra(Config.*TAG\_EDIT*,String.*valueOf*(edit));  
 intent1.putExtra(Config.*TAG\_YEAR*,String.*valueOf*(tmp\_year));  
 intent1.putExtra(Config.*TAG\_MONTH2*,String.*valueOf*(temp\_month));  
 intent1.setFlags(Intent.*FLAG\_ACTIVITY\_CLEAR\_TOP*);  
 startActivity(intent1);  
 return true;  
 case R.id.*next*:  
 if (temp\_month==12){  
 temp\_month = 1;  
 tmp\_year = tmp\_year +1;  
 }else{  
 temp\_month = temp\_month + 1;  
 }  
  
 Intent intent2 = new Intent(this, WriteAttendance.class);  
 intent2.putExtra(Config.*TAG\_EDIT*,String.*valueOf*(edit));  
 intent2.putExtra(Config.*TAG\_YEAR*,String.*valueOf*(tmp\_year));  
 intent2.putExtra(Config.*TAG\_MONTH2*,String.*valueOf*(temp\_month));  
 intent2.setFlags(Intent.*FLAG\_ACTIVITY\_CLEAR\_TOP*);  
 startActivity(intent2);  
 return true;  
  
 case R.id.*menu\_item\_edit*: //this item has your app icon  
 Intent intent = new Intent(WriteAttendance.this, AttendedMonth.class);  
 intent.putExtra(Config.*TAG\_DATE*,date);  
 intent.putExtra(Config.*TAG\_EDIT*, edit);  
 intent.putExtra(Config.*TAG\_MONTH*, month);  
 intent.putExtra(Config.*TAG\_YEAR*, year);  
 intent.putExtra(Config.*KEY\_CUSTOMER\_NAME*, customer\_name);  
 intent.putExtra(Config.*KEY\_PROJECT\_NAME*, project\_name);  
 intent.putExtra(Config.*KEY\_WO\_NUMBER*, wo\_number);  
 intent.putExtra(Config.*TAG\_TOTALTIME*,totaltime);  
 intent.putExtra(Config.*TAG\_OVERTIME*,overtime);  
 startActivity(intent);  
 return true;  
  
 default: return super.onOptionsItemSelected(item);  
 }  
}

### Item list



File Java : CustomSimpleAdapter.java

File xml : item\_list.xml

In WriteAttendance.java the function of showDay() will send params into CustomSimpleAdapter.java

private void showDay(){  
 JSONObject jsonObject = null;  
 ArrayList<HashMap<String,String>> list = new ArrayList<HashMap<String, String>>();  
 try {  
 jsonObject = new JSONObject(JSON\_STRING);  
 JSONArray result = jsonObject.getJSONArray("data");  
 String status\_fetch = jsonObject.getString(Config.*TAG\_STATUS*);  
 if (status\_fetch.equals("200")){  
 date = result.getJSONObject(0).getString(Config.*TAG\_DATE*);  
 for(int i = 0; i<result.length(); i++){  
 JSONObject jo = result.getJSONObject(i);  
 String day = jo.getString(Config.*TAG\_DAY\_NAME*);  
 String date = jo.getString(Config.*TAG\_DATE*);  
 String status = jo.getString(Config.*TAG\_STATUS*);  
 String date2 = jo.getString(Config.*TAG\_DATE2*);  
 String time=null;  
 String time\_in = "";  
 String time\_out = "";  
 String time\_break = "";  
 String place = "";  
 String activity = "";  
 String overtime="", totaltime="";  
 String information=null;  
 if (status.equals("1")) {  
 time = jo.getString(Config.*TAG\_TIME\_IN*)+" - "+jo.getString(Config.*TAG\_TIME\_OUT*);  
 information = jo.getString(Config.*TAG\_PLACE*)+" ("+jo.getString(Config.*TAG\_ACTIVITY*)+")";  
 time\_in = jo.getString(Config.*TAG\_TIME\_IN*);  
 time\_out = jo.getString(Config.*TAG\_TIME\_OUT*);  
 time\_break = jo.getString(Config.*TAG\_TIME\_BREAK*);  
 place = jo.getString(Config.*TAG\_PLACE*);  
 overtime = jo.getString(Config.*TAG\_OVERTIME*);  
 totaltime = jo.getString(Config.*TAG\_TOTALTIME*);  
 activity = jo.getString(Config.*TAG\_ACTIVITY*);  
 }else{  
 time = "";  
 information = "";  
 }  
  
 HashMap<String,String> dataDay = new HashMap<>();  
 dataDay.put(Config.*TAG\_DAY\_NAME*,day);  
 dataDay.put(Config.*TAG\_DATE*,date);  
 dataDay.put(Config.*TAG\_DATE2*,date2);  
 dataDay.put(Config.*TAG\_TIME\_IN\_OUT*,time);  
 dataDay.put(Config.*TAG\_TIME\_IN*,time\_in);  
 dataDay.put(Config.*TAG\_TIME\_OUT*,time\_out);  
 dataDay.put(Config.*TAG\_TIME\_BREAK*,time\_break);  
 dataDay.put(Config.*TAG\_PLACE*,place);  
 dataDay.put(Config.*TAG\_ACTIVITY*,activity);  
 dataDay.put(Config.*TAG\_TIME\_INFORMATION*,information);  
 dataDay.put(Config.*TAG\_STATUS*,status);  
 dataDay.put(Config.*TAG\_TOTALTIME*,totaltime);  
 dataDay.put(Config.*TAG\_OVERTIME*,overtime);  
 list.add(dataDay);  
 }  
 }else if (status\_fetch.equals("401")){  
 String message = jsonObject.getString(Config.*TAG\_MESSAGE*);  
 Toast.*makeText*(WriteAttendance.this,message,Toast.*LENGTH\_LONG*).show();  
 Logout();  
 }  
  
 } catch (JSONException e) {  
 e.printStackTrace();  
 }  
  
 SimpleAdapter adapter2 = new CustomSimpleAdapter(this, list,  
 R.layout.*item\_list*, new String[] { Config.*TAG\_DAY\_NAME*,Config.*TAG\_DATE*,Config.*TAG\_TIME\_IN\_OUT*,Config.*TAG\_TIME\_INFORMATION*,Config.*TAG\_STATUS*}, new  
 int[] {R.id.*day*, R.id.*date*, R.id.*time\_in\_out*,R.id.*time\_information*,R.id.*day*});  
  
 listView.setAdapter(adapter2);  
}

To give action if item clicked in function onItemClick() in WriteAttendance.java

@Override  
public void onItemClick(AdapterView<?> parent, View view, int position, long id) {  
  
 Intent intent = new Intent(this, AttendedDay.class);  
  
 HashMap<String,String> map =(HashMap)parent.getItemAtPosition(position);  
 String date2 = map.get(Config.*TAG\_DATE2*).toString();  
 String date = map.get(Config.*TAG\_DATE*).toString();  
 String status = map.get(Config.*TAG\_STATUS*).toString();  
 String time\_in = map.get(Config.*TAG\_TIME\_IN*).toString();  
 String time\_out = map.get(Config.*TAG\_TIME\_OUT*).toString();  
 String time\_break = map.get(Config.*TAG\_TIME\_BREAK*).toString();  
 String activity = map.get(Config.*TAG\_ACTIVITY*).toString();  
 String place = map.get(Config.*TAG\_PLACE*).toString();  
 String overtime = map.get(Config.*TAG\_OVERTIME*).toString();  
 String totaltime = map.get(Config.*TAG\_TOTALTIME*).toString();  
  
 intent.putExtra(Config.*TAG\_DATE*,date);  
 intent.putExtra(Config.*TAG\_DATE2*,date2);  
 intent.putExtra(Config.*TAG\_TIME\_IN*,time\_in);  
 intent.putExtra(Config.*TAG\_TIME\_OUT*,time\_out);  
 intent.putExtra(Config.*TAG\_TIME\_BREAK*,time\_break);  
 intent.putExtra(Config.*TAG\_STATUS*,status);  
 intent.putExtra(Config.*TAG\_PLACE*,place);  
 intent.putExtra(Config.*TAG\_ACTIVITY*,activity);  
 intent.putExtra(Config.*TAG\_TOTALTIME*,totaltime);  
 intent.putExtra(Config.*TAG\_OVERTIME*,overtime);  
 intent.putExtra(Config.*TAG\_EDIT*,edit);  
 intent.putExtra(Config.*TAG\_MONTH2*,month);  
 intent.putExtra(Config.*TAG\_YEAR*,year);  
 startActivity(intent);  
}

This action will open **AttendedDay.class** (daily attendance form).

To set background color of item list in **CustomSimpleAdapter.class**

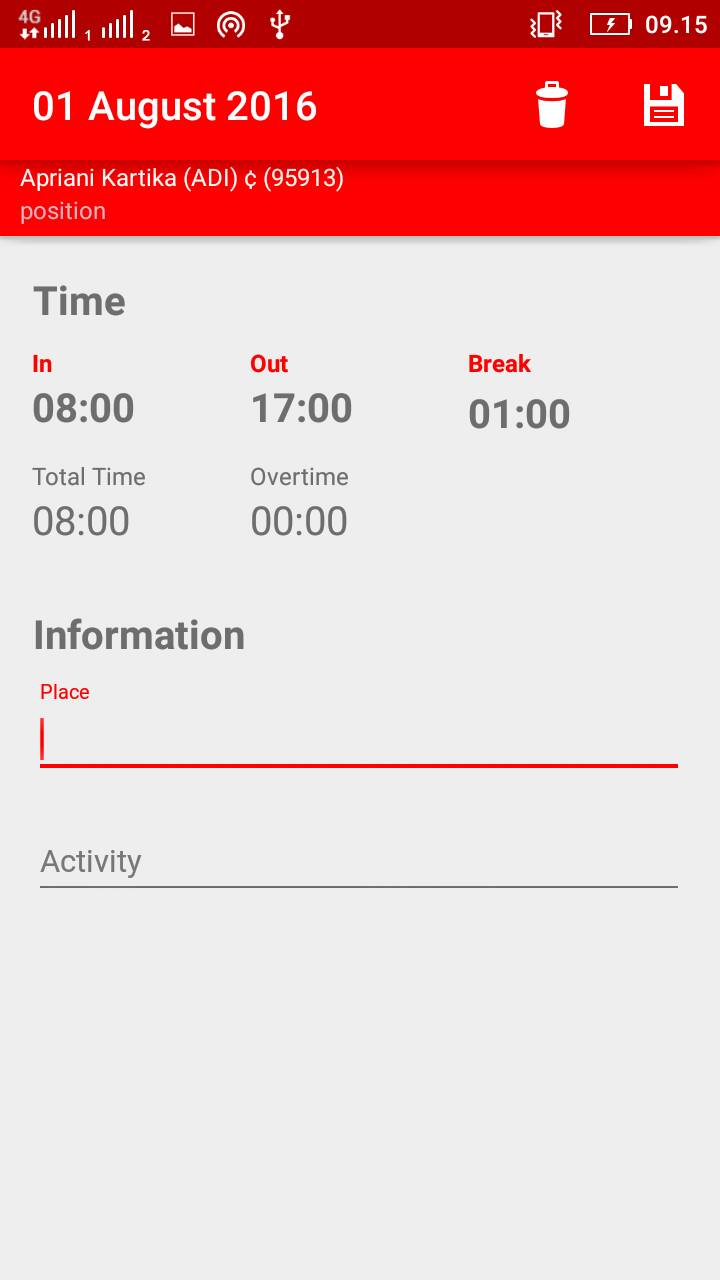
holder.day.setText((CharSequence) itemList.get(position).get(Config.*TAG\_DAY\_NAME*));  
holder.date.setText((CharSequence) itemList.get(position).get(Config.*TAG\_DATE*));  
holder.time\_information.setText((CharSequence) itemList.get(position).get(Config.*TAG\_TIME\_INFORMATION*));  
holder.time\_in\_out.setText((CharSequence) itemList.get(position).get(Config.*TAG\_TIME\_IN\_OUT*));  
String status = ((String)itemList.get(position).get(Config.*TAG\_STATUS*));  
if (status.equals("1")){  
 holder.day.setBackgroundResource(R.drawable.*bg\_gray*);  
 holder.day.setTextColor(Color.*parseColor*("#FFFFFF"));  
 holder.time\_in\_out.setTextSize(16);  
 holder.time\_information.setTextSize(16);  
}else if (status.equals("2")){  
 holder.day.setBackgroundResource(R.drawable.*bg\_red*);  
 holder.day.setTextColor(Color.*parseColor*("#FFFFFF"));  
 holder.time\_in\_out.setTextSize(0);  
 holder.time\_information.setTextSize(0);  
}else if (status.equals("0")){  
 holder.day.setBackgroundResource(R.drawable.*bg\_white*);  
 holder.day.setTextColor(Color.*DKGRAY*);  
 holder.time\_in\_out.setTextSize(0);  
 holder.time\_information.setTextSize(0);  
}

status = 1 means the day was attended by employee and the background color will be grey( file in **res\drawable\bg\_gray.xml** ).

Status = 2 means the day is holiday and the background color will be red (file in **res\drawable\bg\_red.xml** ).

Status = 0 means the day not yet be attended by employee and the background color will be white (file in **res\drawable\bg\_white.xml** ).

## Daily Attendance Form



File java : **AttendedDay.java**

File xml : **activity\_attended\_day.xml**

This class will recieve several parameter

private void getDataIntent(){  
 Intent intent = getIntent();  
 i\_date2 = intent.getStringExtra(Config.*TAG\_DATE2*);  
 i\_date = intent.getStringExtra(Config.*TAG\_DATE*);  
 timed\_in = intent.getStringExtra(Config.*TAG\_TIME\_IN*);  
 timed\_out = intent.getStringExtra(Config.*TAG\_TIME\_OUT*);  
 timed\_break = intent.getStringExtra(Config.*TAG\_TIME\_BREAK*);  
 i\_activity = intent.getStringExtra(Config.*TAG\_ACTIVITY*);  
 i\_place = intent.getStringExtra(Config.*TAG\_PLACE*);  
 i\_overtime = intent.getStringExtra(Config.*TAG\_OVERTIME*);  
 i\_totaltime = intent.getStringExtra(Config.*TAG\_TOTALTIME*);  
 status = intent.getStringExtra(Config.*TAG\_STATUS*);  
 edit = intent.getStringExtra(Config.*TAG\_EDIT*);  
 month = intent.getStringExtra(Config.*TAG\_MONTH2*);  
 year = intent.getStringExtra(Config.*TAG\_YEAR*);  
}

* i\_date2 is use to post information of the day will be attended (ex value: 2016/08/01)
* i\_date is use to set title of toolbar (ex value: 01 August 2016)
* timed\_in use to set ‘time in’ in the form of daily attendance
* timed\_out use to set’ time out’ in the form of daily attendance
* timed\_break use to set’ time break’ in the form of daily attendance
* i\_activity use to set’ activity’ in the form of daily attendance
* place use to set’ place’ in the form of daily attendance
* i\_overtime use to set’ overtime’ in the form of daily attendance
* I\_totaltime use to set’ totaltime’ in the form of daily attendance
* Status = 1 means the record of the day is not empty
* Edit = 1 mean the form can be edited, edit = 0 mean the form can’t be edited (just view)

### Set default time in, out, break

File in **AttendedDay.java**

if (edit.equals("1")){  
 setContentView(R.layout.*activity\_attended\_day*);  
 initialization();  
 activity = (AutoCompleteTextView) findViewById(R.id.*activity*);  
 place = (AutoCompleteTextView) findViewById(R.id.*place*);  
 if (status.equals("1")){  
 activity.setText(i\_activity);  
 place.setText(i\_place);  
 setValue();  
 }else{  
 timed\_in="08:00";  
 timed\_out="17:00";  
 timed\_break="01:00";  
 }

default time in : 08:00

default time out : 17:00

default time break : 01:00

### C:\Users\Ryan Baskara\AppData\Local\Microsoft\Windows\INetCache\Content.Word\dailyAttendanceForm.jpgToolbar icon

File in **res\menu\attended\_day.xml**

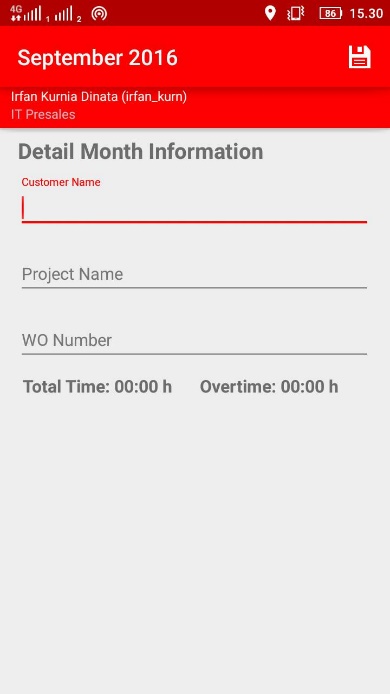
To set action of icon in **AttendedDay.java**

@Override  
public boolean onOptionsItemSelected(MenuItem item){  
  
 switch(item.getItemId()){  
 case R.id.*menu\_delete*: //this item has your app icon  
 deleteAttendedDay = new DeleteAttendedDay();  
 deleteAttendedDay.execute();  
 return true;  
  
 case R.id.*menu\_save*:  
 attendedDaySubmit();  
 return true;  
  
 default: return super.onOptionsItemSelected(item);  
 }  
}

### Delete Record daily attendance

public class DeleteAttendedDay extends AsyncTask<Void, Void, Boolean> {  
 ProgressDialog loading;  
 DeleteAttendedDay() {  
  
 }  
  
 protected void onPreExecute() {  
 super.onPreExecute();  
 loading = ProgressDialog.*show*(AttendedDay.this,null,"Deleting record",false,true);  
 }

## Monthly Information Form



File java : **AttendedMonth.java**

File XML : **res\layout\activity\_attended\_month.xml**

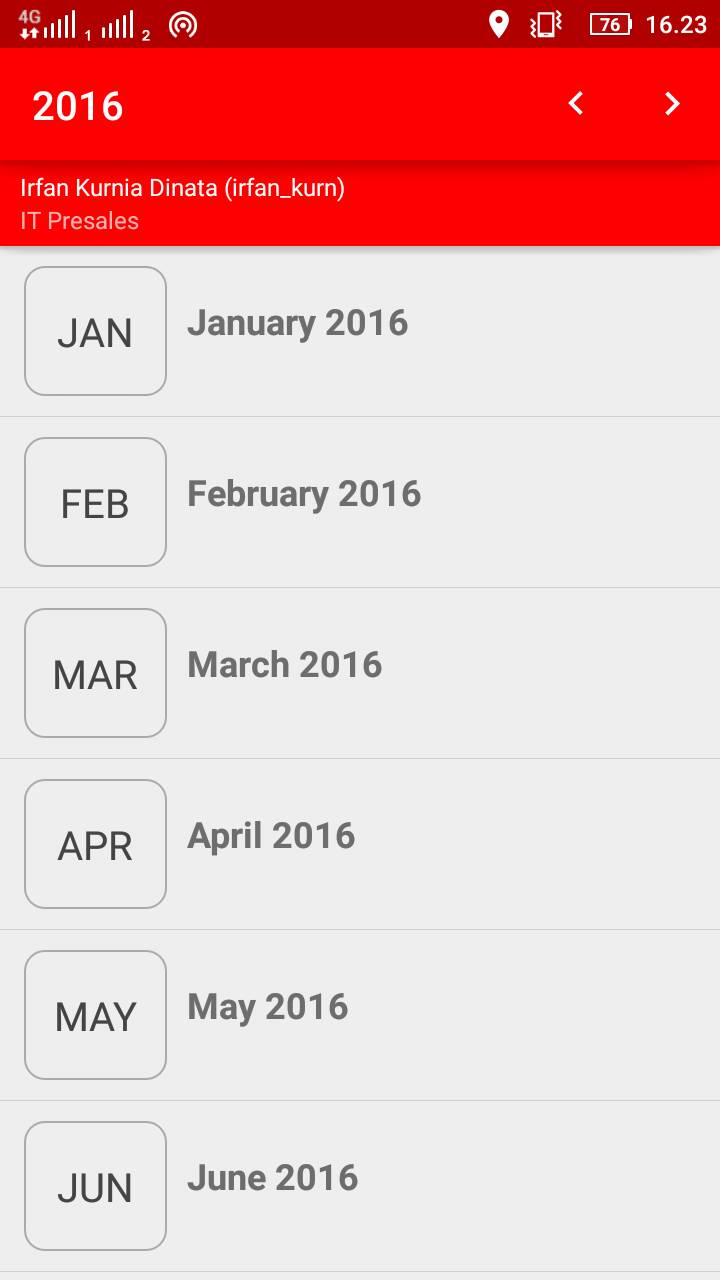
### Set required to all field

public void saveInformation(){  
 View focusView = null;  
  
 custName = customer\_name.getText().toString();  
 pName = project\_name.getText().toString();  
 woNumber = wo\_number.getText().toString();  
  
 if (TextUtils.*isEmpty*(custName)){  
 customer\_name.setError(getString(R.string.*error\_field\_required*));  
 focusView = customer\_name;  
 focusView.requestFocus();  
 }  
 if (TextUtils.*isEmpty*(pName)){  
 project\_name.setError(getString(R.string.*error\_field\_required*));  
 focusView = project\_name;  
 focusView.requestFocus();  
 }  
 if (TextUtils.*isEmpty*(woNumber)){  
 wo\_number.setError(getString(R.string.*error\_field\_required*));  
 focusView = wo\_number;  
 focusView.requestFocus();  
 }else {  
 postInformation = new PostInformation();  
 postInformation.execute((Void) null);  
 }  
  
}

### Post information to server

@Override  
protected Boolean doInBackground(Void... params) {  
 // *TODO: attempt authentication against a network service.* try {  
 HashMap<String,String> param = new HashMap<>();  
 param.put(Config.*TAG\_TOKEN*,token);  
 param.put(Config.*TAG\_ID*,id);  
 param.put(Config.*KEY\_CUSTOMER\_NAME*,custName);  
 param.put(Config.*KEY\_PROJECT\_NAME*,pName);  
 param.put(Config.*KEY\_WO\_NUMBER*,woNumber);  
 param.put(Config.*TAG\_DATE*,year+"-"+month+"-01");  
 RequestHandler rh = new RequestHandler();  
 String res = rh.sendPostRequest(Config.*URL\_POST\_MONTH*, param);  
 JSON\_STRING = res;  
 Thread.*sleep*(3000);  
 } catch (InterruptedException e) {  
 return false;  
 }  
  
 return true;  
}

## View Attendance



File java : **ViewAttendance.java**

File xml : **res\layout\activity\_list.xml**

### Get list month

private void showMonth(){  
 JSONObject jsonObject = null;  
 ArrayList<HashMap<String,String>> list = new ArrayList<HashMap<String, String>>();  
 try {  
 jsonObject = new JSONObject(JSON\_STRING);  
 JSONArray result = jsonObject.getJSONArray("data");  
 String status = jsonObject.getString(Config.*TAG\_STATUS*);  
 if (status.equals("200")) {  
 for (int i = 0; i < result.length(); i++) {  
 JSONObject jo = result.getJSONObject(i);  
 String month = jo.getString(Config.*TAG\_MONTH*);  
 String month\_name = jo.getString(Config.*TAG\_MONTH2*);  
 String date = jo.getString(Config.*TAG\_DATE*);  
 String status\_month = jo.getString(Config.*TAG\_STATUS*);  
 String attended = jo.getString(Config.*TAG\_ATTENDED*);  
  
 HashMap<String, String> dataMonth = new HashMap<>();  
 dataMonth.put(Config.*TAG\_MONTH*, month);  
 dataMonth.put(Config.*TAG\_NAME*, month\_name);  
 dataMonth.put(Config.*TAG\_DATE*, date);  
 dataMonth.put(Config.*TAG\_STATUS*,status\_month);  
 dataMonth.put(Config.*TAG\_ATTENDED*,"Attended: "+attended+" days");  
 String customer\_name=null, project\_name=null, wo\_number=null,totaltime=null;  
 if (status\_month.equals("1")){  
 customer\_name = jo.getString(Config.*KEY\_CUSTOMER\_NAME*);  
 project\_name = jo.getString(Config.*KEY\_PROJECT\_NAME*);  
 wo\_number = jo.getString(Config.*KEY\_WO\_NUMBER*);  
 String overtime = jo.getString(Config.*TAG\_OVERTIME*);  
 totaltime ="Total Time: " +jo.getString(Config.*TAG\_TOTALTIME*)+" (Overtime: "+overtime+")";  
 }  
  
 dataMonth.put(Config.*KEY\_CUSTOMER\_NAME*, customer\_name);  
 dataMonth.put(Config.*KEY\_PROJECT\_NAME*, project\_name);  
 dataMonth.put(Config.*KEY\_WO\_NUMBER*, wo\_number);  
 dataMonth.put(Config.*TAG\_TIME\_INFORMATION*, totaltime);  
 list.add(dataMonth);  
 }  
 }else if(status.equals("401")){  
 String message = jsonObject.getString(Config.*TAG\_MESSAGE*);  
 Toast.*makeText*(ViewAttendance.this,message,Toast.*LENGTH\_LONG*).show();  
 Logout();  
 }  
 } catch (JSONException e) {  
 e.printStackTrace();  
 }  
  
 SimpleAdapter adapter2 = new Adapter\_ViewAttendance(this, list,  
 R.layout.*item\_list*, new String[] { Config.*TAG\_NAME*,Config.*TAG\_DATE*,Config.*TAG\_ATTENDED*,Config.*KEY\_CUSTOMER\_NAME*,Config.*KEY\_PROJECT\_NAME*,Config.*KEY\_WO\_NUMBER*,Config.*TAG\_TIME\_INFORMATION*,Config.*TAG\_STATUS*}, new  
 int[] {R.id.*name*, R.id.*date*,R.id.*attended*,R.id.*time*,R.id.*customer\_name*,R.id.*project\_name*,R.id.*wo\_number* });  
  
 listView.setAdapter(adapter2);  
}

item list for month using Adapter\_ViewAttendance.java to custom view each item

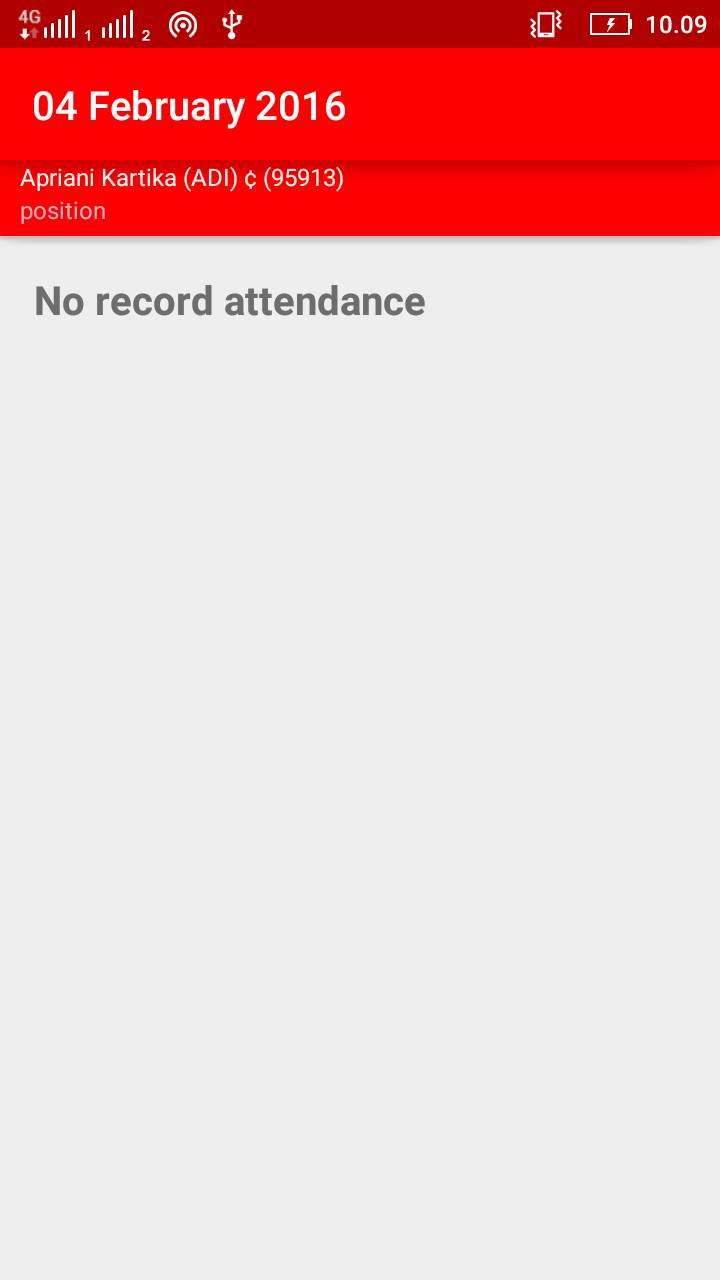
### Get list day

This function use WriteAttendance.java but set TAG EDIT into “0”

@Override  
public void onItemClick(AdapterView<?> parent, View view, int position, long id) {  
 Intent intent = new Intent(this, WriteAttendance.class);  
 HashMap<String,String> map =(HashMap)parent.getItemAtPosition(position);  
 String month = map.get(Config.*TAG\_MONTH*).toString();  
  
 intent.putExtra(Config.*TAG\_MONTH2*,month);  
 intent.putExtra(Config.*TAG\_YEAR*,String.*valueOf*(year));  
 intent.putExtra(Config.*TAG\_EDIT*,"0");  
 intent.setFlags(Intent.*FLAG\_ACTIVITY\_CLEAR\_TOP*);  
 startActivity(intent);  
}

### Disabled record of daily attendance

### No record of daily attendance



File xml : **res\layout\activity\_attended\_day\_disabled\_no\_record.xml**

<TextView  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:textSize="20dp"  
 android:paddingLeft="16dp"  
 android:text="@string/no\_record"  
 android:textStyle="bold"  
 android:gravity="left"  
 android:layout\_column="0"  
 android:layout\_span="6" />

To set text in r**es\values\strings.xml**

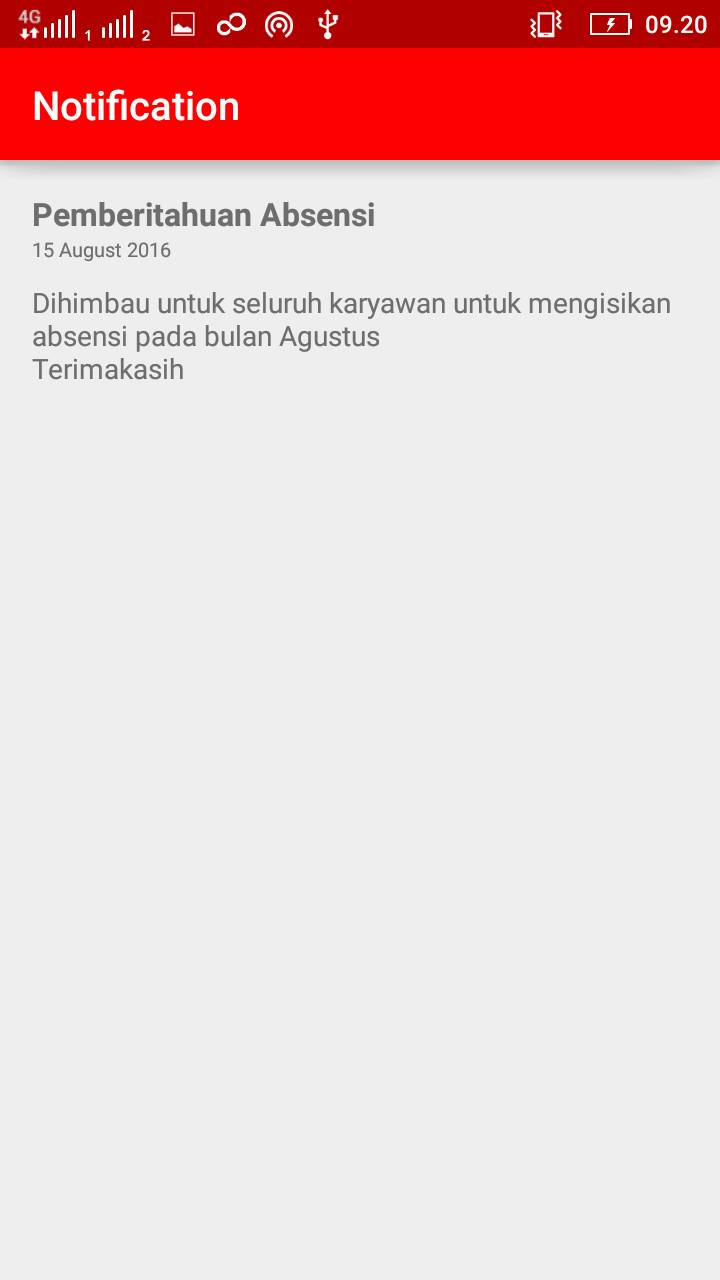
<string name="no\_record">No record attendance</string>

## Notification

File java : **GCMPushReceiverService.java, GCMRegistrationIntentService.java, GCMTokenRefreshListenerService.java**

### Notification page view

File java : Notification.java



File xml : activity\_notification.xml

### Show notification

File java : **GCMPushReceiverService.java**

private void sendNotification(String message) {  
 String title=null,content=null,timestamp=null,date=null;  
 JSONObject jsonObject = null;  
 try {  
 jsonObject = new JSONObject(message);  
 title = jsonObject.getString(Config.*TAG\_TITLE*);  
 content = jsonObject.getString(Config.*TAG\_CONTENT*);  
 date = jsonObject.getString(Config.*TAG\_DATE*);  
  
 } catch (JSONException e) {  
 e.printStackTrace();  
 }

json object is used to parsing json into object.

Tittle used to set notification tittle.

Content used to set notification content.

Date used to show date notification was pushed.

SharedPreferences sharedPreferences;  
 sharedPreferences = getSharedPreferences("Notification", Context.*MODE\_PRIVATE*);  
 SharedPreferences.Editor editor = sharedPreferences.edit();  
 editor.putString(Config.*TAG\_TITLE*,title);  
 editor.putString(Config.*TAG\_CONTENT*,content);  
 editor.putString(Config.*TAG\_DATE*,date);  
 editor.apply();

SharePreference used to save notification data into local storage so if you clear data in mobile working report apps, the notification page in the apps will be empty.

Intent intent = new Intent(this, Notification.class);  
 intent.addFlags(Intent.*FLAG\_ACTIVITY\_CLEAR\_TOP*);  
  
 int requestCode = 0;  
 PendingIntent pendingIntent = PendingIntent.*getActivity*(this, requestCode, intent, PendingIntent.*FLAG\_ONE\_SHOT*);  
 Uri sound = RingtoneManager.*getDefaultUri*(RingtoneManager.*TYPE\_NOTIFICATION*);  
  
 Bitmap bitmap = BitmapFactory.*decodeResource*(getResources(),R.drawable.*notification\_logo*);

Bitmap is used to display large icon notification, src : **res\drawable\notification\_logo.png**

NotificationCompat.Builder noBuilder = new NotificationCompat.Builder(this)  
 .setSmallIcon(R.drawable.*just\_logo\_red*)  
 .setLargeIcon(bitmap)  
 .setContentTitle(title)  
 .setContentText(content)  
 .setStyle(new NotificationCompat.BigTextStyle().bigText(content))  
 .setSound(sound)  
 .setAutoCancel(true)  
 .setContentIntent(pendingIntent);  
  
 NotificationManager notificationManager = (NotificationManager)getSystemService(Context.*NOTIFICATION\_SERVICE*);  
 notificationManager.notify(0, noBuilder.build()); //0 = ID of notification  
}

Src small icon logo in **res\drawable\just\_logo\_red.png**