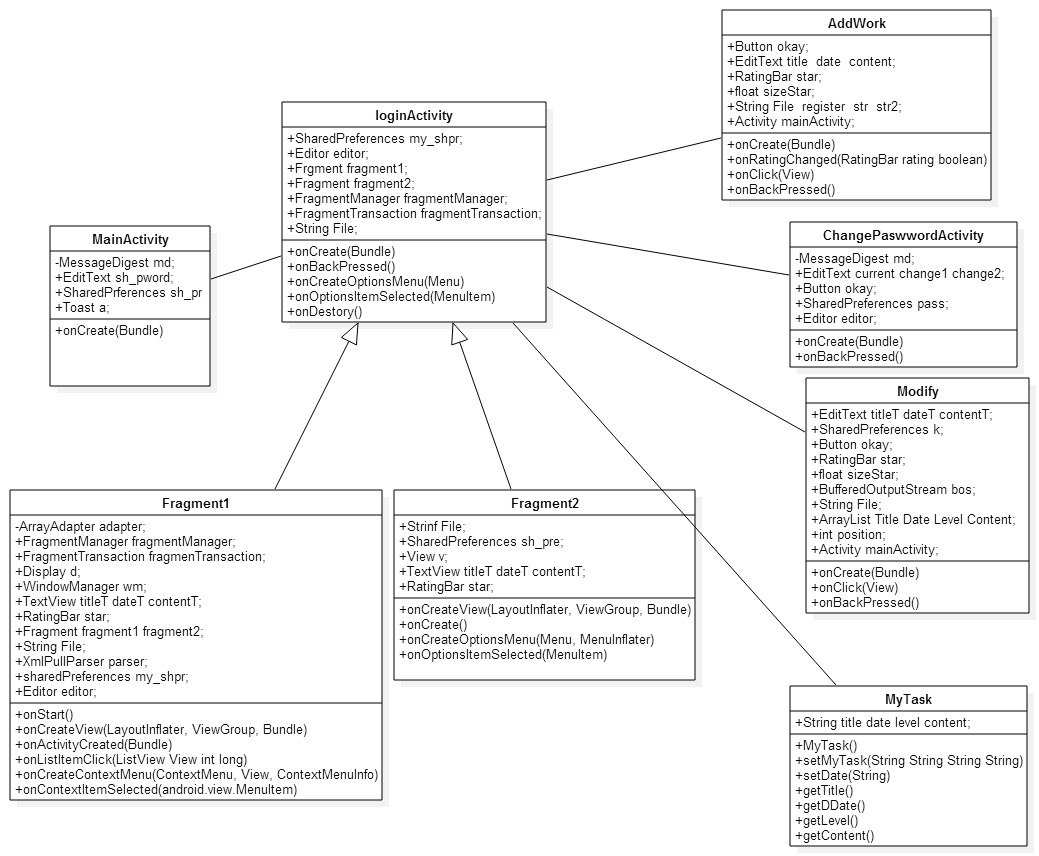
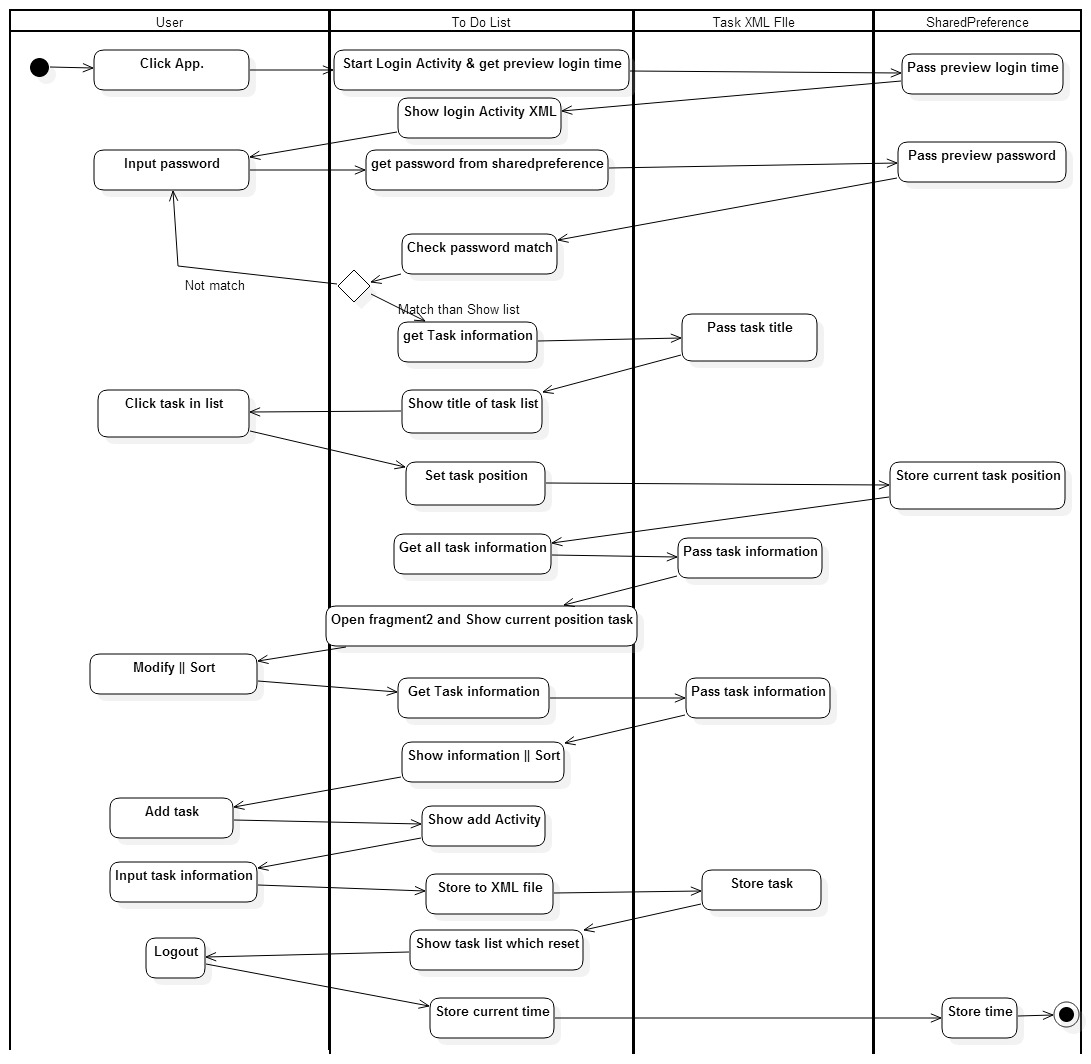
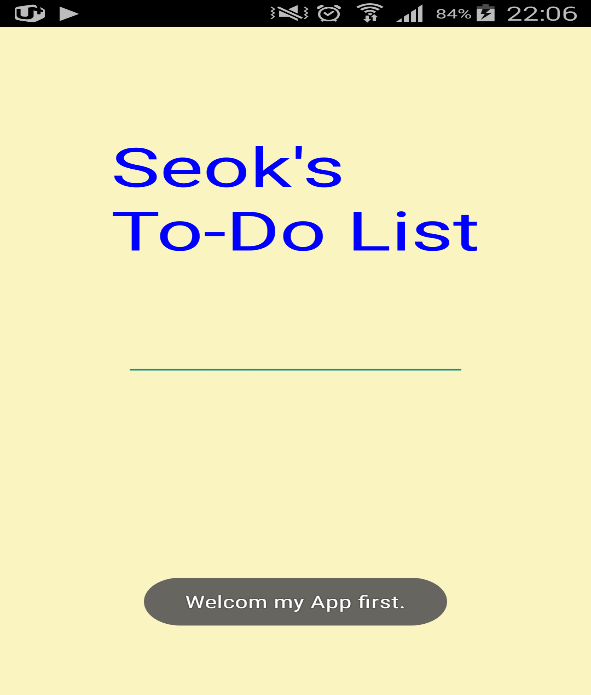
**Homework 4**

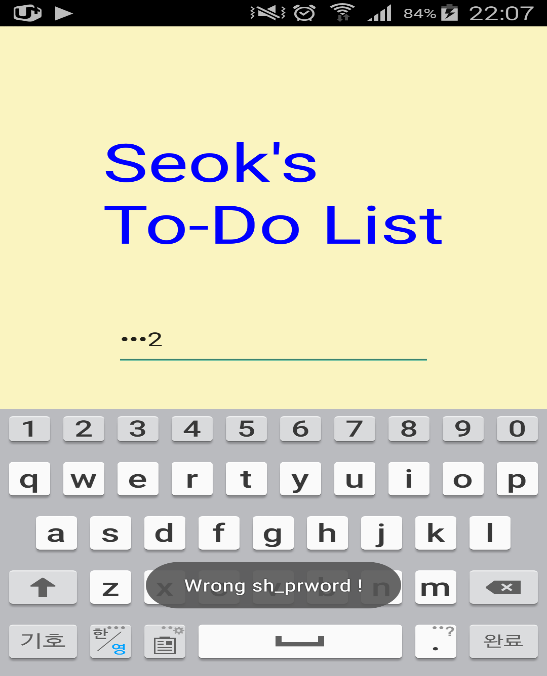
**201133216 정유석**

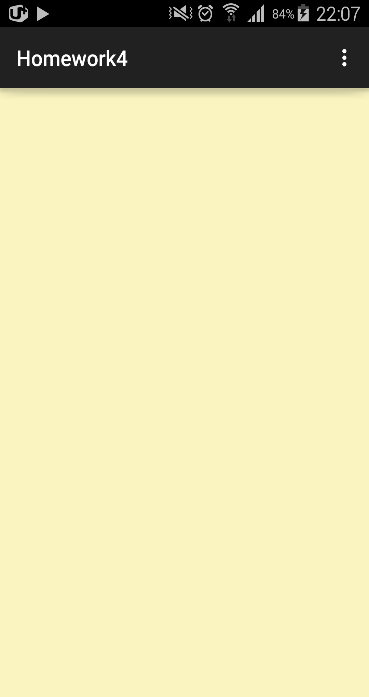
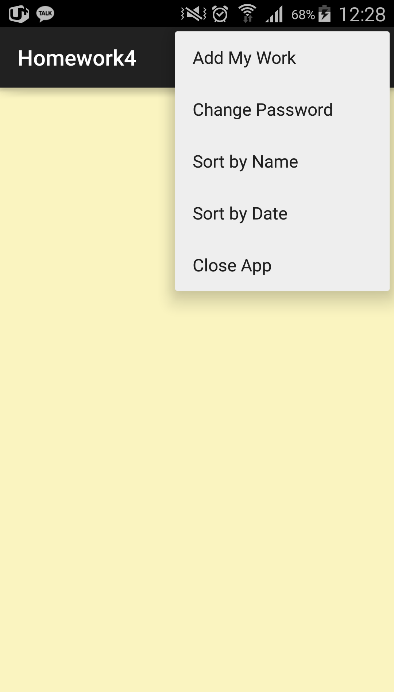
**Schema Of My Program**

****

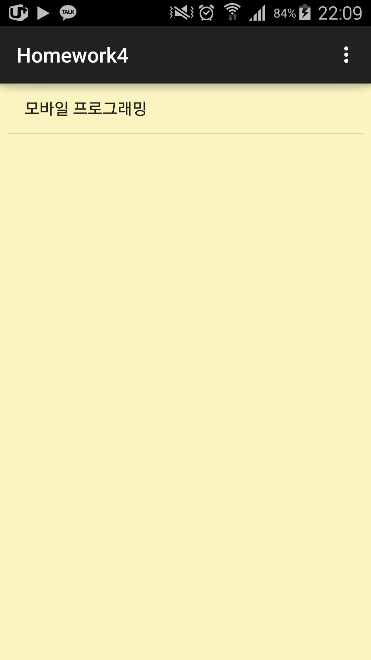
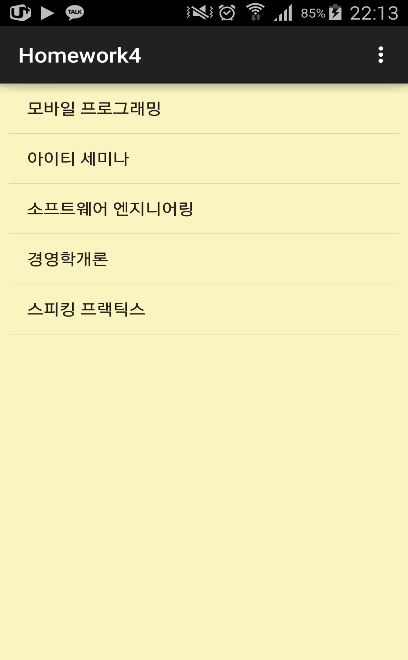
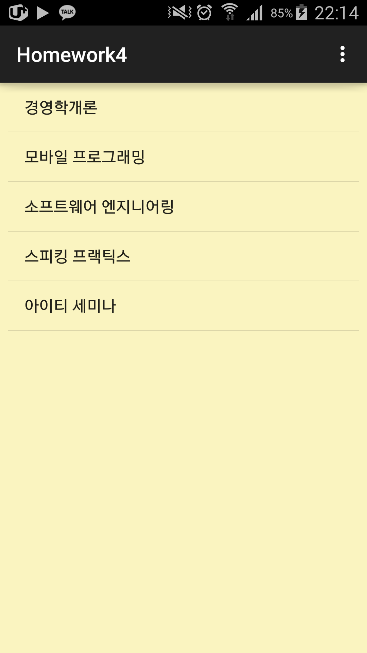
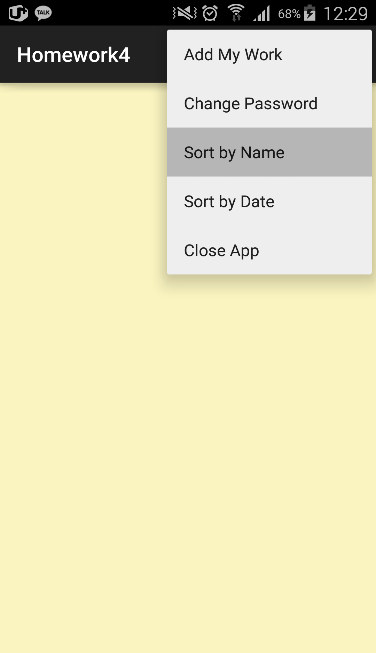
****

**ScreenShot**

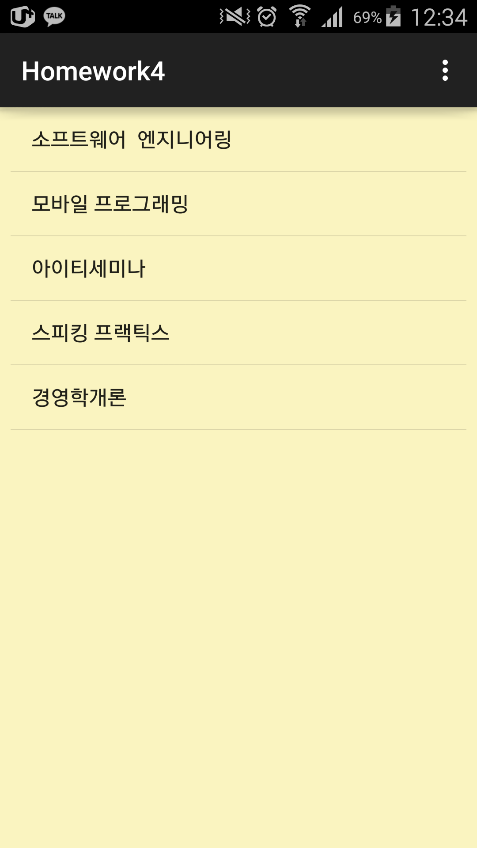
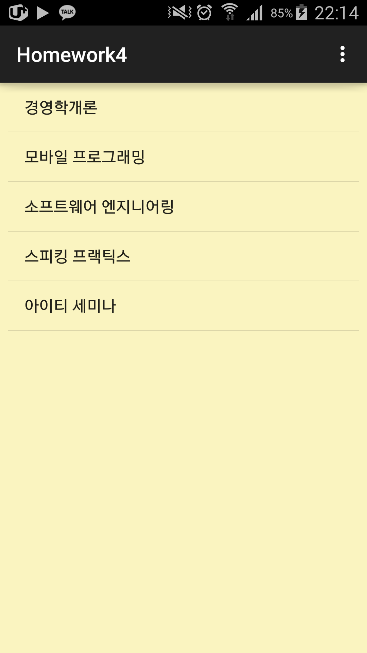


****  **<Login Display>**

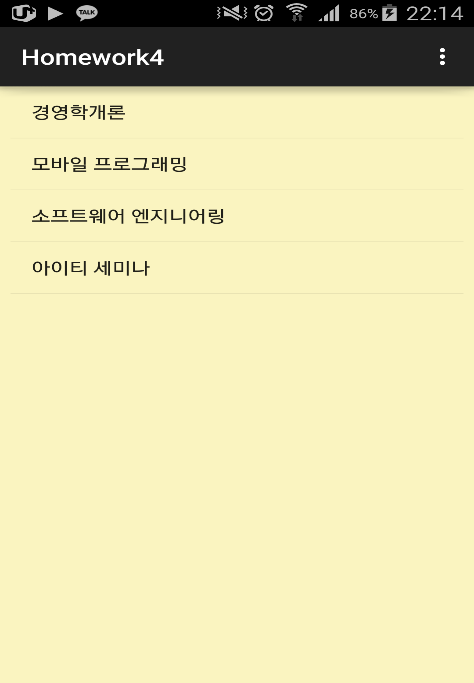
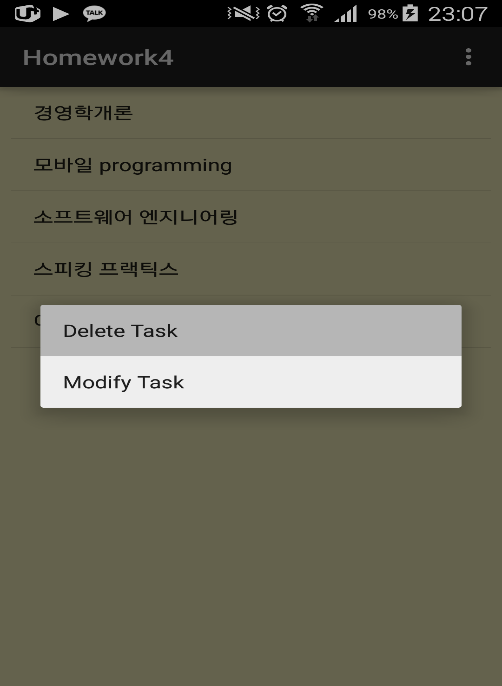
**<First List Display> <Add Task Display>**

****

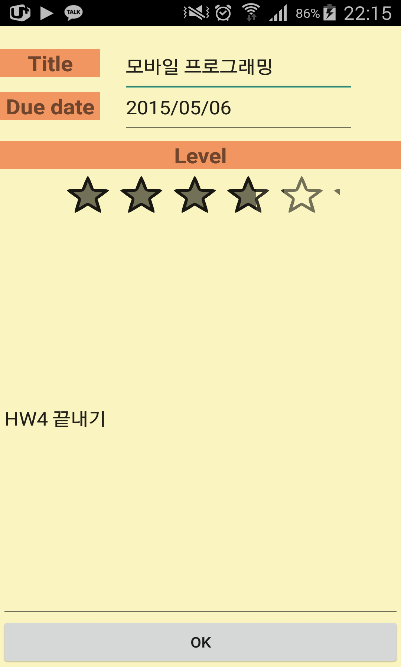
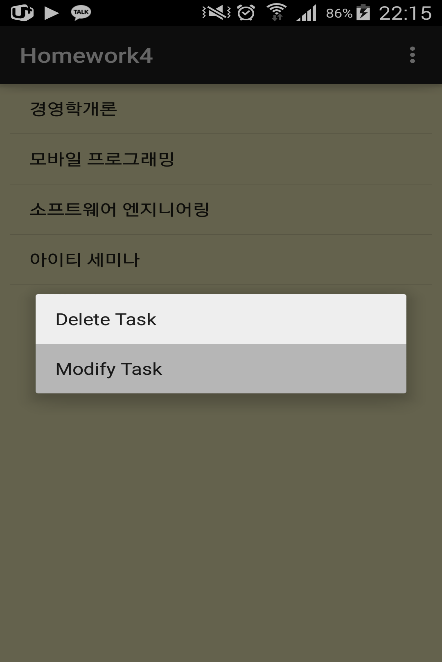
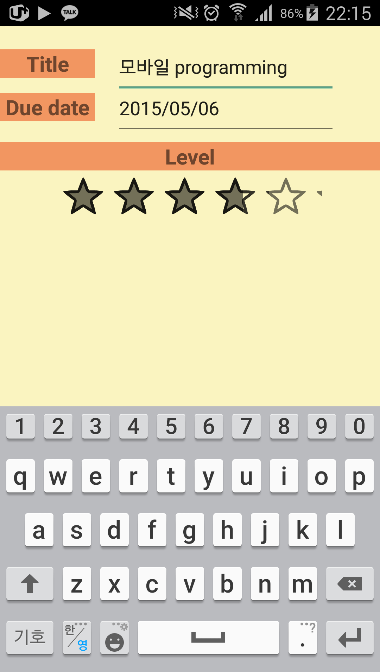
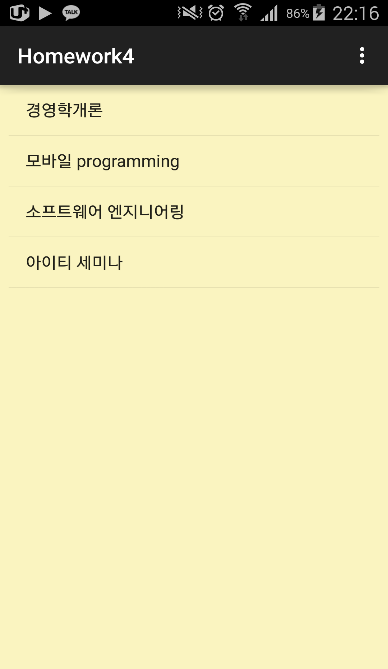
**<Sorting Task By Name>**

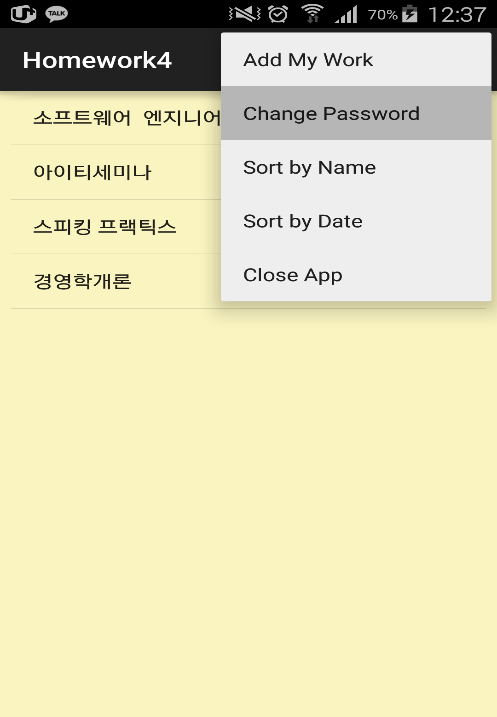
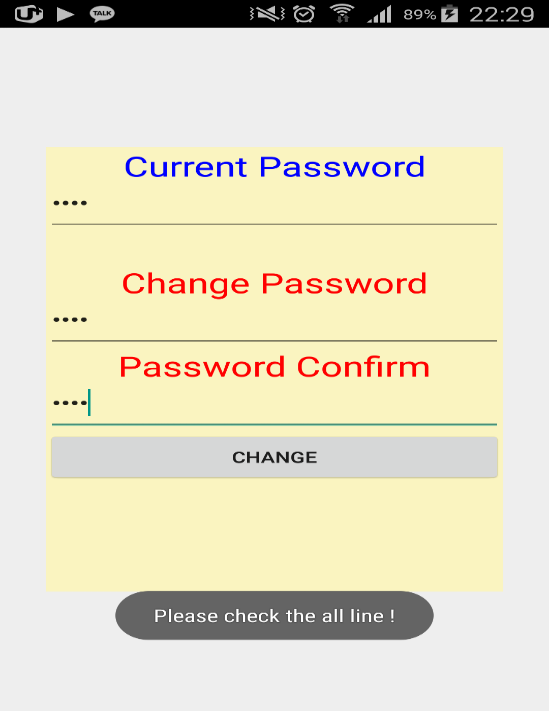
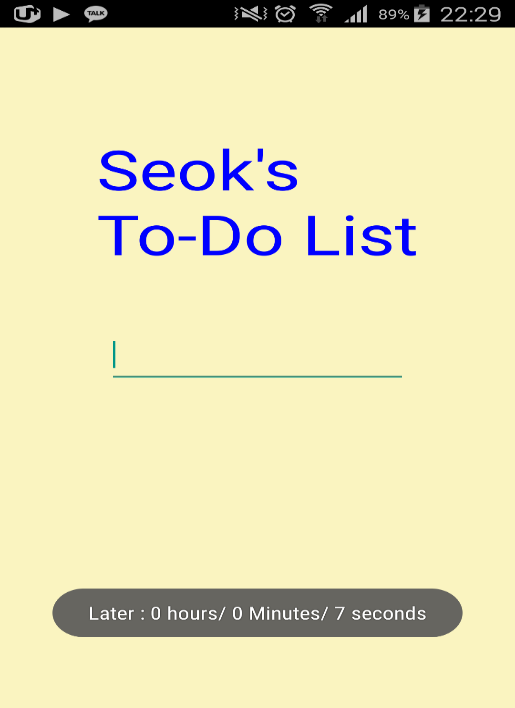
****

**<Sorting Task By Date>**

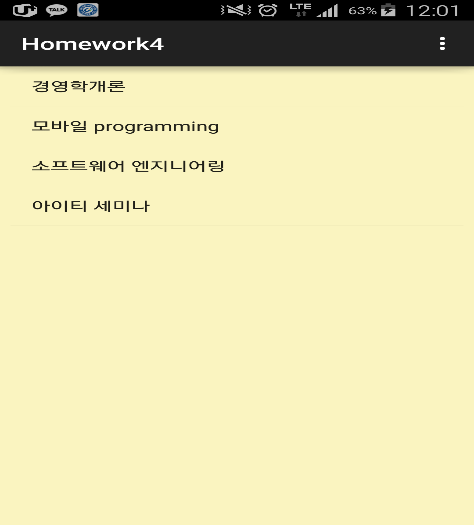
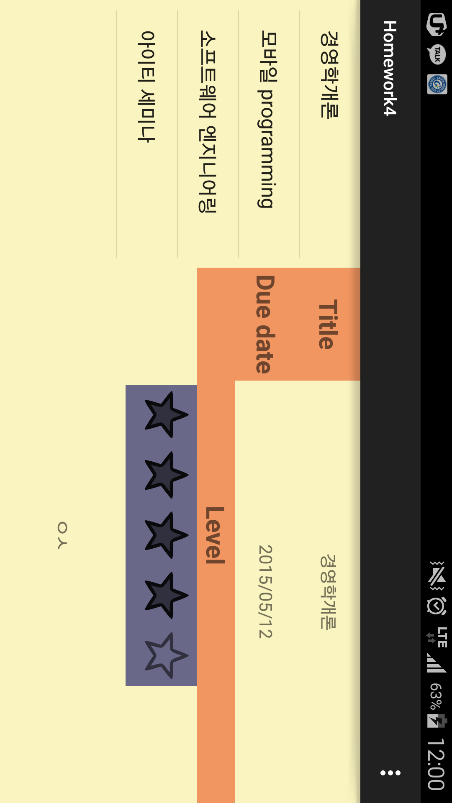
****

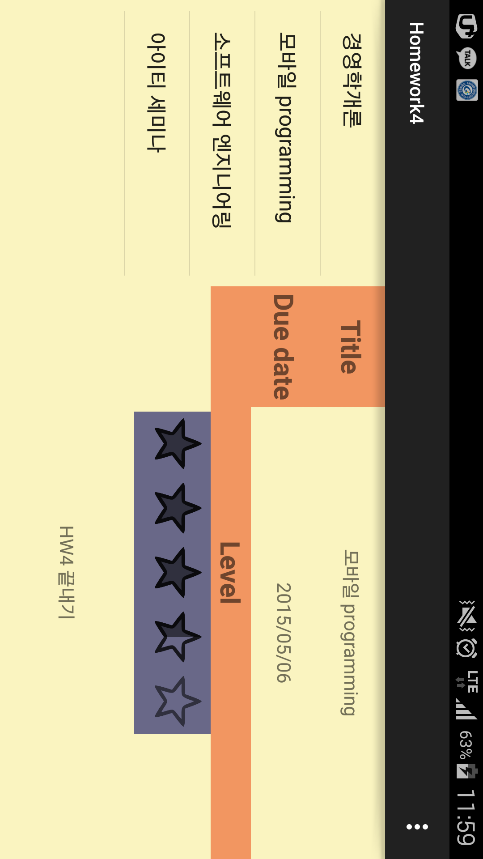
**<Context Menu of Delete>**

** <Context Menu of Modify>**

****

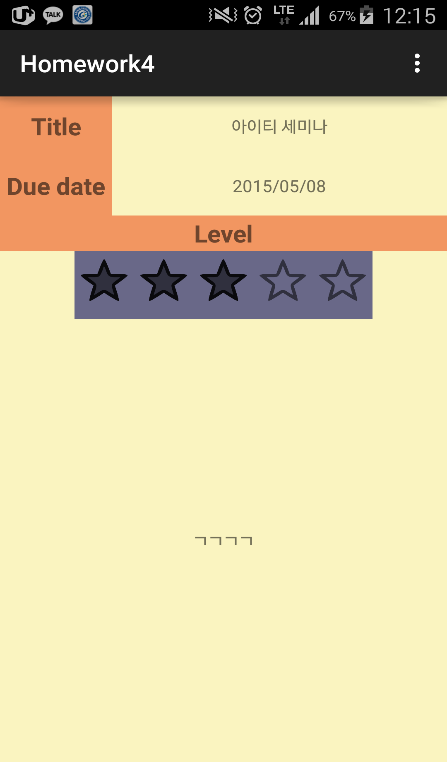
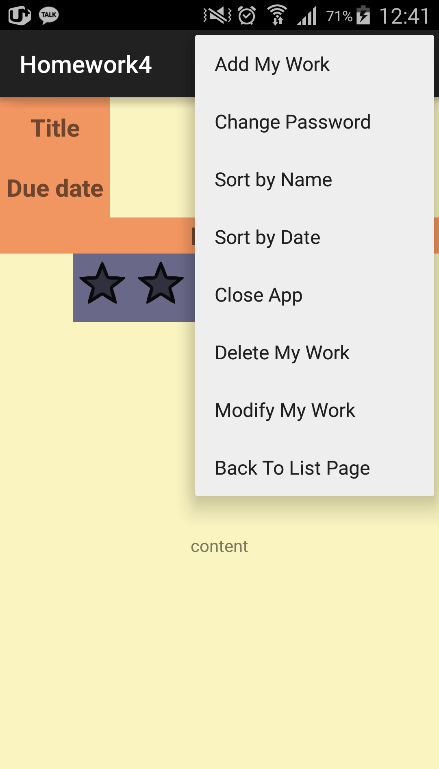
**<Change Password & Reaccess App>**

****

****

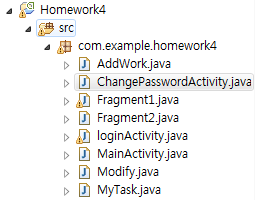
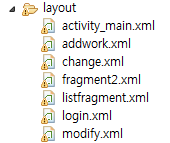
****

**<Land Scape Mode>**

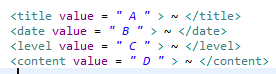
****

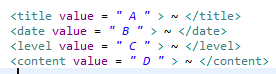
**<Option Menu in Task>**

**My JAVA Source & XML File**





**My XML Shape & App Style**



**CODE**

**<Login Activity = ‘MainActivity’ (Java source name)>**

**package com.example.homework4;**

**import java.security.MessageDigest;**

**import java.security.NoSuchAlgorithmException;**

**import android.app.Activity;**

**import android.content.Intent;**

**import android.content.SharedPreferences;**

**import android.os.Bundle;**

**import android.text.Editable;**

**import android.text.TextWatcher;**

**import android.widget.EditText;**

**import android.widget.Toast;**

**//At first, we have to give our secret code (4 digits) to start the App**

**public class MainActivity extends Activity {**

**private static MessageDigest md; //Using compare password in preference**

**EditText sh\_prword; //getting user input password**

**SharedPreferences sh\_pr; //get password information**

**Toast a;**

**@Override**

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

**super.onCreate(savedInstanceState);**

**setContentView(R.layout.login);**

**//It shows this sentence when user inputs wrong password**

**a = Toast.makeText(this, "Wrong sh\_prword !", Toast.LENGTH\_SHORT);**

**//It is used to get password from shared preferences**

**sh\_pr = getSharedPreferences("My ToDo-list",0);**

**sh\_prword = (EditText)findViewById(R.id.editText1);**

**//It is used to check date of preview access**

**long dateCheck = sh\_pr.getLong("Date", -1);**

**if(dateCheck == -1){ //First come to App**

**Toast.makeText(this,"Welcom my App first.", Toast.LENGTH\_SHORT).show();**

**}else{ //Else**

**long now = System.currentTimeMillis();**

**now = (now - dateCheck) / 1000; //Now - Preview time using Millis seconds**

**int totalSecond = (int)now;**

**int hour, minute,second;**

**hour = totalSecond / 3600;**

**minute = totalSecond % 3600 / 60;**

**second = totalSecond % 3600 % 60;**

**//Print time**

**String time = "Later : " + hour + " hours/ " + minute + " Minutes/ " + second+" seconds";**

**Toast.makeText(this,time, Toast.LENGTH\_SHORT).show();**

**}**

**//The password consists of 4 digits. So, when 4-th digit is given by the user,**

**//the following password testing process should start automatically without explicitly**

**//pressing a [OK] button**

**sh\_prword.addTextChangedListener(new TextWatcher() {**

**@Override**

**public void onTextChanged(CharSequence s, int start, int before, int count) {**

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

**}**

**@Override**

**public void beforeTextChanged(CharSequence s, int start, int count,**

**int after) {**

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

**}**

**@Override**

**public void afterTextChanged(Editable s) {**

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

**if(s.toString().length() == 4){ //Check the password number**

**String check = sh\_pr.getString("password","-1");**

**String pass = s.toString();**

**String compare = null ;**

**try { // password encryption, The password kept in storage with encrypted format**

**md = MessageDigest.getInstance("MD5"); //Using MD5**

**byte[] passBytes = pass.getBytes(); //Change bytes**

**md.reset();**

**byte[] digested = md.digest(passBytes); //Encryption**

**StringBuffer sb = new StringBuffer();**

**for(int i=0;i<digested.length;i++)**

**sb.append(Integer.toHexString(0xff & digested[i]));**

**compare = sb.toString();**

**} catch (NoSuchAlgorithmException e) {**

**// TODO Auto-generated catch block**

**e.printStackTrace();**

**}**

**if(check.equals("-1")){ //If it first access, Compare '0000'**

**if(s.toString().equals("0000")){**

**startActivity(new Intent("com.example.homework4.loginActivity"));**

**finish();**

**}**

**else**

**a.show();**

**}**

**else if(check.equals(compare)){ //If it is same password in preference**

**startActivity(new Intent("com.example.homework4.loginActivity"));**

**finish();**

**}**

**else**

**a.show();**

**}**

**else if(s.toString().length() > 4)**

**a.show();**

**}**

**});**

**}**

**}**

**<Show list task Activity = ‘loginActivity’> (it has 2 fragments)**

**package com.example.homework4;**

**import java.io.FileOutputStream;**

**import java.io.IOException;**

**import java.io.InputStream;**

**import java.io.OutputStreamWriter;**

**import java.text.ParseException;**

**import java.text.SimpleDateFormat;**

**import java.util.ArrayList;**

**import java.util.Collections;**

**import java.util.Comparator;**

**import org.xmlpull.v1.XmlPullParser;**

**import org.xmlpull.v1.XmlPullParserException;**

**import org.xmlpull.v1.XmlPullParserFactory;**

**import android.support.v7.app.ActionBarActivity;**

**import android.app.Fragment;**

**import android.app.FragmentManager;**

**import android.app.FragmentTransaction;**

**import android.content.Intent;**

**import android.content.SharedPreferences;**

**import android.os.Bundle;**

**import android.util.Log;**

**import android.view.Display;**

**import android.view.Menu;**

**import android.view.MenuItem;**

**import android.view.WindowManager;**

**import android.widget.Toast;**

**//It is main board of App. So it has two fragment(list fragment, fragment)**

**public class loginActivity extends ActionBarActivity {**

**SharedPreferences my\_shpr ; //Set task position (position is used to delete or modify task**

**SharedPreferences.Editor editor;**

**Fragment fragment1; //Get fragment id**

**Fragment fragment2;**

**FragmentManager fragmentManager; //it manages fragments**

**FragmentTransaction fragmentTransaction;**

**String File = "czc.xml"; //File name**

**@Override**

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

**super.onCreate(savedInstanceState);**

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

**//Set shared preferences**

**my\_shpr = getSharedPreferences("My ToDo-list",0);**

**editor = my\_shpr.edit();**

**//It is used to fragment management**

**fragmentManager = getFragmentManager();**

**fragmentTransaction = fragmentManager.beginTransaction();**

**//it gets each fragment id**

**fragment1 = fragmentManager.findFragmentById(R.id.Fragment1);**

**fragment2 = fragmentManager.findFragmentById(R.id.Fragment2);**

**//---get the current display info---**

**WindowManager wm= getWindowManager();**

**Display d = wm.getDefaultDisplay();**

**//if height larger than width, App only shows list fragment using 'show','hide'**

**if (d.getHeight() > d.getWidth()){**

**fragmentTransaction.hide(fragment2);**

**fragmentTransaction.commit();**

**}**

**}**

**//if user press back button in fragment2 (Task fragment), App shows list fragment1 using 'show', 'hide'**

**@Override**

**public void onBackPressed(){**

**if(fragment1.isHidden()){**

**fragmentManager = getFragmentManager();**

**fragmentTransaction = fragmentManager.beginTransaction();**

**fragmentTransaction.hide(fragment2);**

**fragmentTransaction.show(fragment1);**

**fragmentTransaction.commit();**

**}**

**else**

**finish();**

**}**

**@Override**

**public boolean onCreateOptionsMenu(Menu menu) {**

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

**//The app is controlled by an OPTION MENU offering the following 5 possibilities:**

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

**return true;**

**}**

**//It handles option event**

**@Override**

**public boolean onOptionsItemSelected(MenuItem item) {**

**//get id from user event**

**int id = item.getItemId();**

**//if Add, start 'AddWork' activity**

**if(id == R.id.Add){**

**startActivity(new Intent("com.example.homework4.AddWork"));**

**finish();**

**}**

**//if change, start 'ChangePasswordActivity' activity**

**if (id == R.id.change){**

**startActivity(new Intent("com.example.homework4.ChangePasswordActivity"));**

**finish();**

**}**

**//if Close, put now date to preference and close App**

**if(id == R.id.Close){**

**editor.putLong("Date", System.currentTimeMillis());**

**editor.commit();**

**finish();**

**}**

**//if Sort, App sorts all task using arrayList and MyTask object (sort by name)**

**if (id == R.id.sortByName){**

**//They using sort by title or date**

**ArrayList<MyTask> Task = new ArrayList(); //Task has information of task (title, level..)**

**try{ //Get information from XML folder**

**InputStream In = openFileInput(File);**

**if(In != null){ //Using XML parsing**

**XmlPullParserFactory xf = XmlPullParserFactory.newInstance(); //create XmlPullparserFactory**

**XmlPullParser parser = xf.newPullParser();**

**// XmlPullParser에 XML 데이터와 인코딩 방식을 입력**

**parser.setInput(In, "utf-8");**

**// custom\_list.xml 을 가져와 XmlPullParser 에 넣는다.**

**// 파싱한 xml 이 END\_DOCUMENT(종료 태그)가 나올때 까지 반복한다.**

**int size = 0; //It is used to check if 'parser' gave one task information**

**String TTitle="", DDate="", LLevel="", CContent="";**

**while(parser.getEventType()!=XmlPullParser.END\_DOCUMENT){**

**// 태그의 첫번째 속성일 때,**

**if(parser.getEventType()==XmlPullParser.START\_TAG){**

**// 이름이 "custom" 일때, 첫번째 속성값을 ArrayList에 저장**

**if(parser.getName().equals("title")){**

**TTitle = parser.getAttributeValue(0); //put title**

**size ++; //size up**

**}**

**if(parser.getName().equals("date")){**

**DDate = (parser.getAttributeValue(0)); //put date**

**size ++;**

**}**

**if(parser.getName().equals("level")){**

**LLevel = parser.getAttributeValue(0); //put level**

**size ++;**

**}**

**if(parser.getName().equals("content")){**

**CContent = parser.getAttributeValue(0); //put content**

**size ++;**

**}**

**if(size % 4 == 0){**

**//if size = 4, parser gave one task information perfectly**

**MyTask putData = new MyTask(); //put object**

**putData.setMyTask(TTitle, DDate, LLevel, CContent);**

**//It is used to sort task after**

**Task.add(putData);**

**//initialize one task information**

**TTitle = "";**

**DDate = "";**

**LLevel = "";**

**CContent = "";**

**}**

**}**

**// move next tag**

**parser.next();**

**}**

**}**

**In.close();**

**}catch (IllegalArgumentException e) {**

**// TODO Auto-generated catch block**

**e.printStackTrace();**

**} catch (IllegalStateException e) {**

**// TODO Auto-generated catch block**

**e.printStackTrace();**

**} catch (IOException e) {**

**// TODO Auto-generated catch block**

**e.printStackTrace();**

**} catch (XmlPullParserException e) {**

**// TODO Auto-generated catch block**

**e.printStackTrace();**

**}**

**//Sorting task**

**Collections.sort(Task, new Comparator<MyTask>(){**

**@Override**

**public int compare(MyTask o1, MyTask o2){ //compare title**

**return o1.title.compareTo(o2.title);**

**}**

**});**

**//put data to xml file**

**try {**

**String register = "";**

**FileOutputStream fOut = openFileOutput(File,MODE\_PRIVATE); //open file**

**OutputStreamWriter osw= new OutputStreamWriter(fOut); //create writer**

**for(MyTask task : Task){**

**//put all task data using XML shape**

**register = register + "<title value = \"" + task.getTitle() + "\">" +"</title>";**

**register = register + "\n" + "<date value = \"" + task.getDDate() + "\">" + "</date>";**

**register = register + "\n" + "<content value = \"" + task.getContent() + "\">" + "</content>";**

**register = register + "\n" + "<level value = \"" + task.getLevel() + "\">" + "</level>";**

**}**

**osw.write(register); //write the file**

**osw.close(); //close file**

**}catch (IllegalArgumentException e) {**

**// TODO Auto-generated catch block**

**e.printStackTrace();**

**}catch (IllegalStateException e) {**

**// TODO Auto-generated catch block**

**e.printStackTrace();**

**} catch (IOException e) {**

**// TODO Auto-generated catch block**

**e.printStackTrace();**

**}**

**//restart activity**

**startActivity(new Intent("com.example.homework4.loginActivity"));**

**finish();**

**}**

**//if Sort, App sorts all task using arrayList and MyTask object (sort by date)**

**if (id == R.id.sortByDate){**

**//They using sort by title or date**

**ArrayList<MyTask> Task2 = new ArrayList(); //Task has information of task (title, level..)**

**try{ //Get information from XML folder**

**InputStream In = openFileInput(File);**

**if(In != null){ //Using XML parsing**

**XmlPullParserFactory xf = XmlPullParserFactory.newInstance(); //create XmlPullparserFactory**

**XmlPullParser parser = xf.newPullParser();**

**// XmlPullParser에 XML 데이터와 인코딩 방식을 입력**

**parser.setInput(In, "utf-8");**

**// custom\_list.xml 을 가져와 XmlPullParser 에 넣는다.**

**// 파싱한 xml 이 END\_DOCUMENT(종료 태그)가 나올때 까지 반복한다.**

**int size = 0; //It is used to check if 'parser' gave one task information**

**String TTitle="", DDate="", LLevel="", CContent="";**

**while(parser.getEventType()!=XmlPullParser.END\_DOCUMENT){**

**// 태그의 첫번째 속성일 때,**

**if(parser.getEventType()==XmlPullParser.START\_TAG){**

**// 이름이 "custom" 일때, 첫번째 속성값을 ArrayList에 저장**

**if(parser.getName().equals("title")){**

**TTitle = parser.getAttributeValue(0); //put title**

**size ++; //size up**

**}**

**if(parser.getName().equals("date")){**

**DDate = (parser.getAttributeValue(0)); //put date**

**size ++;**

**}**

**if(parser.getName().equals("level")){**

**LLevel = parser.getAttributeValue(0); //put level**

**size ++;**

**}**

**if(parser.getName().equals("content")){**

**CContent = parser.getAttributeValue(0); //put content**

**size ++;**

**}**

**if(size % 4 == 0){**

**//if size = 4, parser gave one task information perfectly**

**MyTask putData = new MyTask(); //put object**

**putData.setMyTask(TTitle, DDate, LLevel, CContent);**

**//It is used to sort task after**

**Task2.add(putData);**

**//initialize one task information**

**TTitle = "";**

**DDate = "";**

**LLevel = "";**

**CContent = "";**

**}**

**}**

**// move next tag**

**parser.next();**

**}**

**}**

**In.close();**

**}catch (IllegalArgumentException e) {**

**// TODO Auto-generated catch block**

**e.printStackTrace();**

**} catch (IllegalStateException e) {**

**// TODO Auto-generated catch block**

**e.printStackTrace();**

**} catch (IOException e) {**

**// TODO Auto-generated catch block**

**e.printStackTrace();**

**} catch (XmlPullParserException e) {**

**// TODO Auto-generated catch block**

**e.printStackTrace();**

**}**

**//Sorting task**

**Collections.sort(Task2, new Comparator<MyTask>(){**

**@Override**

**public int compare(MyTask o1, MyTask o2){ //compare date**

**return o1.date.compareTo(o2.date);**

**}**

**});**

**//put data to xml file**

**try {**

**String register = "";**

**FileOutputStream fOut = openFileOutput(File,MODE\_PRIVATE); //open file**

**OutputStreamWriter osw= new OutputStreamWriter(fOut); //create writer**

**for(MyTask task : Task2){**

**//put all task data using XML shape**

**register = register + "<title value = \"" + task.getTitle() + "\">" +"</title>";**

**register = register + "\n" + "<date value = \"" + task.getDDate() + "\">" + "</date>";**

**register = register + "\n" + "<content value = \"" + task.getContent() + "\">" + "</content>";**

**register = register + "\n" + "<level value = \"" + task.getLevel() + "\">" + "</level>";**

**}**

**osw.write(register); //write the file**

**osw.close(); //close file**

**}catch (IllegalArgumentException e) {**

**// TODO Auto-generated catch block**

**e.printStackTrace();**

**}catch (IllegalStateException e) {**

**// TODO Auto-generated catch block**

**e.printStackTrace();**

**} catch (IOException e) {**

**// TODO Auto-generated catch block**

**e.printStackTrace();**

**}**

**//restart activity**

**startActivity(new Intent("com.example.homework4.loginActivity"));**

**finish();**

**}**

**return super.onOptionsItemSelected(item);**

**}**

**//It is used to check date**

**@Override**

**protected void onDestroy() {**

**super.onDestroy(); //if state is destroy, put now date to preference using editor**

**editor.putLong("Date", System.currentTimeMillis()); //Millis shape**

**editor.commit();**

**}**

**}**

**<Show list task fragmet = ‘Fragment1’>**

**package com.example.homework4;**

**import java.io.FileOutputStream;**

**import java.io.IOException;**

**import java.io.InputStream;**

**import java.io.OutputStreamWriter;**

**import java.util.ArrayList;**

**import org.xmlpull.v1.XmlPullParser;**

**import org.xmlpull.v1.XmlPullParserException;**

**import org.xmlpull.v1.XmlPullParserFactory;**

**import android.app.Fragment;**

**import android.app.FragmentManager;**

**import android.app.FragmentTransaction;**

**import android.app.ListFragment;**

**import android.content.Context;**

**import android.content.Intent;**

**import android.content.SharedPreferences;**

**import android.os.Bundle;**

**import android.util.Log;**

**import android.view.ContextMenu;**

**import android.view.ContextMenu.ContextMenuInfo;**

**import android.view.Display;**

**import android.view.LayoutInflater;**

**import android.view.View;**

**import android.view.ViewGroup;**

**import android.view.WindowManager;**

**import android.widget.AdapterView.AdapterContextMenuInfo;**

**import android.widget.ArrayAdapter;**

**import android.widget.ListView;**

**import android.widget.RatingBar;**

**import android.widget.TextView;**

**/\***

**- Use ListView & Adapter for the List of tasks**

**- When a ListItemis selected, show its detailed Task information using a separate Fragment**

**- Show the Context Menu for the task, when there is a long click on a Item [Delete], [Modify]**

**\*/**

**public class Fragment1 extends ListFragment {**

**private ArrayAdapter<String> adapter;**

**FragmentManager fragmentManager; //It manage fragment**

**FragmentTransaction fragmentTransaction; //It is used to show and hide fragment**

**Display d ; //get display information**

**WindowManager wm;**

**TextView titleT, dateT, contentT; //Set text view to fragment2 (when user click list, show fragment2)**

**RatingBar star; //set importance of task**

**Fragment fragment1, fragment2; //get fragment id from XML**

**String File = "czc.xml"; //File name**

**XmlPullParser parser; //Using get data from XML shape (using parser)**

**SharedPreferences my\_shpr ; //Set task position (position is used to delete or modify task**

**SharedPreferences.Editor editor;**

**@Override**

**public void onStart() { //Set each view ID in java from xml**

**super.onStart();**

**titleT = (TextView)getActivity().findViewById(R.id.titleText);**

**dateT = (TextView)getActivity().findViewById(R.id.dueDateText);**

**star = (RatingBar)getActivity().findViewById(R.id.ratingBar1);**

**contentT = (TextView)getActivity().findViewById(R.id.content);**

**}**

**//Show all task lists**

**@Override**

**public View onCreateView(LayoutInflater inflater, ViewGroup container,**

**Bundle savedInstanceState) {**

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

**wm= getActivity().getWindowManager(); //Using fragment manage**

**d = wm.getDefaultDisplay();**

**my\_shpr = getActivity().getSharedPreferences("My ToDo-list",0);**

**editor = my\_shpr.edit();**

**//Using show task list**

**ArrayList<String> titleString= new ArrayList<String>();**

**try{**

**//Open XML file**

**InputStream In = getActivity().openFileInput(File);**

**if(In != null){ //Using parsing**

**//Create XmlPullParserFactory**

**XmlPullParserFactory xf = XmlPullParserFactory.newInstance();**

**XmlPullParser parser = xf.newPullParser();**

**// XmlPullParser에 XML 데이터와 인코딩 방식을 입력**

**parser.setInput(In, "utf-8");**

**// czc.xml 을 가져와 XmlPullParser 에 넣는다.**

**// 파싱한 xml 이 END\_DOCUMENT(종료 태그)가 나올때 까지 반복한다.**

**while(parser.getEventType()!=XmlPullParser.END\_DOCUMENT){**

**// 태그의 첫번째 속성일 때,**

**if(parser.getEventType()==XmlPullParser.START\_TAG){**

**// 이름이 "title" 일때, 첫번째 속성값을 ArrayList에 저장 (<title value="A">)**

**if(parser.getName().equals("title")){**

**titleString.add(parser.getAttributeValue(0));**

**}**

**}**

**// 다음 태그로 이동**

**parser.next();**

**}**

**}**

**}catch (IllegalArgumentException e) {**

**// TODO Auto-generated catch block**

**e.printStackTrace();**

**} catch (IllegalStateException e) {**

**// TODO Auto-generated catch block**

**e.printStackTrace();**

**} catch (IOException e) {**

**// TODO Auto-generated catch block**

**e.printStackTrace();**

**} catch (XmlPullParserException e) {**

**// TODO Auto-generated catch block**

**e.printStackTrace();**

**}**

**//set adapter**

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

**//show list using adapter**

**setListAdapter(adapter);**

**return inflater.inflate(R.layout.listfragment, container, false);**

**}**

**@Override**

**public void onActivityCreated(Bundle savedInstanceState) {**

**//register context menu listener**

**registerForContextMenu(getListView());**

**super.onActivityCreated(savedInstanceState);**

**}**

**//Event listener each list items**

**@Override**

**public void onListItemClick(ListView l, View v, int position, long id) {**

**super.onListItemClick(l, v, position, id);**

**//Get position and store to preference for modify this item**

**editor.putInt("position",position);**

**editor.commit();**

**//Set fragments for transaction**

**fragment1 = getActivity().getFragmentManager().findFragmentById(R.id.Fragment1);**

**fragment2 = getActivity().getFragmentManager().findFragmentById(R.id.Fragment2);**

**//Using set data to fragment2**

**ArrayList<String> Title = new ArrayList<String>();**

**ArrayList<String> Date = new ArrayList<String>();**

**ArrayList<String> Level = new ArrayList<String>();**

**ArrayList<String> Content = new ArrayList<String>();**

**//XML parsing**

**try{**

**InputStream In = getActivity().openFileInput(File);**

**if(In != null){**

**XmlPullParserFactory xf = XmlPullParserFactory.newInstance();**

**XmlPullParser parser = xf.newPullParser();**

**// XmlPullParser에 XML 데이터와 인코딩 방식을 입력**

**parser.setInput(In, "utf-8");**

**// czc.xml 을 가져와 XmlPullParser 에 넣는다.**

**// 파싱한 xml 이 END\_DOCUMENT(종료 태그)가 나올때 까지 반복한다.**

**while(parser.getEventType()!=XmlPullParser.END\_DOCUMENT){**

**// 태그의 첫번째 속성일 때,**

**if(parser.getEventType()==XmlPullParser.START\_TAG){**

**// 이름이 "title" 일때, 첫번째 속성값을 ArrayList에 저장**

**if(parser.getName().equals("title")){**

**Title.add(parser.getAttributeValue(0));**

**}// 이름이 "date" 일때, 첫번째 속성값을 ArrayList에 저장**

**if(parser.getName().equals("date")){**

**Date.add(parser.getAttributeValue(0));**

**}// 이름이 "level" 일때, 첫번째 속성값을 ArrayList에 저장**

**if(parser.getName().equals("level")){**

**Level.add(parser.getAttributeValue(0));**

**}// 이름이 "content" 일때, 첫번째 속성값을 ArrayList에 저장**

**if(parser.getName().equals("content")){**

**Content.add(parser.getAttributeValue(0));**

**}**

**}**

**// 다음 태그로 이동**

**parser.next();**

**}**

**}**

**}catch (IllegalArgumentException e) {**

**// TODO Auto-generated catch block**

**e.printStackTrace();**

**} catch (IllegalStateException e) {**

**// TODO Auto-generated catch block**

**e.printStackTrace();**

**} catch (IOException e) {**

**// TODO Auto-generated catch block**

**e.printStackTrace();**

**} catch (XmlPullParserException e) {**

**// TODO Auto-generated catch block**

**e.printStackTrace();**

**}**

**//Set fragment2 view for user**

**float k = Float.parseFloat(Level.get(position));**

**titleT.setText(Title.get(position));**

**dateT.setText(Date.get(position));**

**contentT.setText(Content.get(position));**

**star.setRating(k); //set rating (level)**

**//if display height is bigger than width, App have to show fragment2 whent list selected in fragment1**

**if (d.getHeight() > d.getWidth()){**

**fragmentManager =getFragmentManager();**

**fragmentTransaction = fragmentManager.beginTransaction();**

**fragmentTransaction.hide(this);**

**fragmentTransaction.show(fragment2);**

**fragmentTransaction.commit();**

**}**

**}**

**//make context menu**

**@Override**

**public void onCreateContextMenu(ContextMenu menu, View v,**

**ContextMenuInfo menuInfo) {**

**super.onCreateContextMenu(menu, v, menuInfo);**

**int delete = 0, Modify = 1;**

**//Make menu items**

**menu.add(0, delete , 0, "Delete Task");**

**menu.add(0, Modify , 0, "Modify Task");**

**}**

**//context click event listener**

**@Override**

**public boolean onContextItemSelected(android.view.MenuItem item) {**

**//It is used to get position of clicked item**

**AdapterContextMenuInfo info = (AdapterContextMenuInfo) item.getMenuInfo();**

**int index = info.position;**

**switch (item.getItemId()) { //If user click delete**

**case 0:**

**//Using set data to fragment2**

**ArrayList<String> Title = new ArrayList<String>();**

**ArrayList<String> Date = new ArrayList<String>();**

**ArrayList<String> Level = new ArrayList<String>();**

**ArrayList<String> Content = new ArrayList<String>();**

**//XML parsing**

**try{**

**InputStream In = getActivity().openFileInput(File);**

**if(In != null){**

**XmlPullParserFactory xf = XmlPullParserFactory.newInstance();**

**XmlPullParser parser = xf.newPullParser();**

**// XmlPullParser에 XML 데이터와 인코딩 방식을 입력**

**parser.setInput(In, "utf-8");**

**// custom\_czc.xml 을 가져와 XmlPullParser 에 넣는다.**

**// 파싱한 xml 이 END\_DOCUMENT(종료 태그)가 나올때 까지 반복한다.**

**while(parser.getEventType()!=XmlPullParser.END\_DOCUMENT){**

**// 태그의 첫번째 속성일 때,**

**if(parser.getEventType()==XmlPullParser.START\_TAG){**

**// 이름이 "title" 일때, 첫번째 속성값을 ArrayList에 저장**

**if(parser.getName().equals("title")){**

**Title.add(parser.getAttributeValue(0));**

**}**

**// 이름이 "date" 일때, 첫번째 속성값을 ArrayList에 저장**

**if(parser.getName().equals("date")){**

**Date.add(parser.getAttributeValue(0));**

**}**

**// 이름이 "level" 일때, 첫번째 속성값을 ArrayList에 저장**

**if(parser.getName().equals("level")){**

**Level.add(parser.getAttributeValue(0));**

**}**

**// 이름이 "content" 일때, 첫번째 속성값을 ArrayList에 저장**

**if(parser.getName().equals("content")){**

**Content.add(parser.getAttributeValue(0));**

**}**

**}**

**// 다음 태그로 이동**

**parser.next();**

**}**

**}**

**}catch (IllegalArgumentException e) {**

**// TODO Auto-generated catch block**

**e.printStackTrace();**

**} catch (IllegalStateException e) {**

**// TODO Auto-generated catch block**

**e.printStackTrace();**

**} catch (IOException e) {**

**// TODO Auto-generated catch block**

**e.printStackTrace();**

**} catch (XmlPullParserException e) {**

**// TODO Auto-generated catch block**

**e.printStackTrace();**

**}**

**//Delete task**

**Title.remove(index);**

**Date.remove(index);**

**Level.remove(index);**

**Content.remove(index);**

**String str2="";**

**//delete task and store now task list to XML file**

**try {**

**//open file and create writer**

**FileOutputStream fOut = getActivity().openFileOutput(File,Context.MODE\_PRIVATE);**

**OutputStreamWriter osw= new OutputStreamWriter(fOut);**

**//set string using xml shape**

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

**str2 = str2 + "\n" + "<title value = \"" + Title.get(i) + "\">" +"</title>";**

**str2 = str2 + "\n" + "<date value = \"" + Date.get(i) + "\">" + "</date>";**

**str2 = str2 + "\n" + "<content value = \"" + Content.get(i) + "\">" + "</content>";**

**str2 = str2 + "\n" + "<level value = \"" + Level.get(i) + "\">" + "</level>";**

**}**

**osw.write(str2); //write XML file**

**osw.close(); //close XML file**

**}catch (IllegalArgumentException e) {**

**// TODO Auto-generated catch block**

**e.printStackTrace();**

**}catch (IllegalStateException e) {**

**// TODO Auto-generated catch block**

**e.printStackTrace();**

**} catch (IOException e) {**

**// TODO Auto-generated catch block**

**e.printStackTrace();**

**}**

**//Restart activity**

**startActivity(new Intent("com.example.homework4.loginActivity"));**

**getActivity().finish();**

**return true;**

**case 1: //Modify task**

**editor.putInt("position", index); //put position to preference using editor**

**editor.commit();**

**//start 'Modify' activity**

**startActivity(new Intent("com.example.homework4.Modify"));**

**getActivity().finish(); //now activity close**

**return true;**

**}**

**return super.onContextItemSelected(item);**

**}**

**}**

**<Show task information fragmet = ‘Fragment2’>**

**package com.example.homework4;**

**import java.io.FileOutputStream;**

**import java.io.IOException;**

**import java.io.InputStream;**

**import java.io.OutputStreamWriter;**

**import java.text.SimpleDateFormat;**

**import java.util.ArrayList;**

**import org.xmlpull.v1.XmlPullParser;**

**import org.xmlpull.v1.XmlPullParserException;**

**import org.xmlpull.v1.XmlPullParserFactory;**

**import android.app.Fragment;**

**import android.app.FragmentManager;**

**import android.app.FragmentTransaction;**

**import android.content.Context;**

**import android.content.Intent;**

**import android.content.SharedPreferences;**

**import android.os.Bundle;**

**import android.util.Log;**

**import android.view.LayoutInflater;**

**import android.view.Menu;**

**import android.view.MenuInflater;**

**import android.view.MenuItem;**

**import android.view.View;**

**import android.view.ViewGroup;**

**import android.widget.RatingBar;**

**import android.widget.TextView;**

**//Show task information from user click event**

**public class Fragment2 extends Fragment {**

**String File = "czc.xml"; //File name**

**SharedPreferences sh\_pr; //Get task position (position is used to delete or modify task**

**View v; //get current fragment layout information**

**TextView titleT, dateT, contentT; //Set task**

**RatingBar star;**

**//make fragment view hierarchy**

**@Override**

**public View onCreateView(LayoutInflater inflater, ViewGroup container,**

**Bundle savedInstanceState) {**

**//get layout information for 'setText'**

**v = inflater.inflate(R.layout.fragment2,container, false);**

**//set preference**

**sh\_pr = getActivity().getSharedPreferences("My ToDo-list",0);**

**//they has task information**

**ArrayList<String> Title = new ArrayList<String>();**

**ArrayList<String> Date = new ArrayList<String>();**

**ArrayList<String> Level = new ArrayList<String>();**

**ArrayList<String> Content = new ArrayList<String>();**

**//get now position**

**int position = sh\_pr.getInt("position", 0);**

**//set each views id to java**

**titleT = (TextView)v.findViewById(R.id.titleText);**

**dateT = (TextView)v.findViewById(R.id.dueDateText);**

**star = (RatingBar)v.findViewById(R.id.ratingBar1);**

**contentT = (TextView)v.findViewById(R.id.content);**

**try{**

**//open file**

**InputStream In = getActivity().openFileInput(File);**

**if(In != null){**

**XmlPullParserFactory xf = XmlPullParserFactory.newInstance();**

**XmlPullParser parser = xf.newPullParser();**

**// XmlPullParser에 XML 데이터와 인코딩 방식을 입력**

**parser.setInput(In, "utf-8");**

**// custom\_czc.xml 을 가져와 XmlPullParser 에 넣는다.**

**// 파싱한 xml 이 END\_DOCUMENT(종료 태그)가 나올때 까지 반복한다.**

**while(parser.getEventType()!=XmlPullParser.END\_DOCUMENT){**

**// 태그의 첫번째 속성일 때,**

**if(parser.getEventType()==XmlPullParser.START\_TAG){**

**// 이름이 "title" 일때, 첫번째 속성값을 ArrayList에 저장**

**if(parser.getName().equals("title")){**

**Title.add(parser.getAttributeValue(0));**

**}**

**// 이름이 "date" 일때, 첫번째 속성값을 ArrayList에 저장**

**if(parser.getName().equals("date")){**

**Date.add(parser.getAttributeValue(0));**

**}**

**// 이름이 "level" 일때, 첫번째 속성값을 ArrayList에 저장**

**if(parser.getName().equals("level")){**

**Level.add(parser.getAttributeValue(0));**

**}**

**// 이름이 "content" 일때, 첫번째 속성값을 ArrayList에 저장**

**if(parser.getName().equals("content")){**

**Content.add(parser.getAttributeValue(0));**

**}**

**}**

**// 다음 태그로 이동**

**parser.next();**

**}**

**In.close();**

**}**

**}catch (IllegalArgumentException e) {**

**// TODO Auto-generated catch block**

**e.printStackTrace();**

**} catch (IllegalStateException e) {**

**// TODO Auto-generated catch block**

**e.printStackTrace();**

**} catch (IOException e) {**

**// TODO Auto-generated catch block**

**e.printStackTrace();**

**} catch (XmlPullParserException e) {**

**// TODO Auto-generated catch block**

**e.printStackTrace();**

**}**

**if(Title.isEmpty()){**

**return v; //If XML hasn't task, show empty space**

**}else{**

**//If XML has tasks, show current task information**

**float k = Float.parseFloat(Level.get(position));**

**titleT.setText(Title.get(position));**

**dateT.setText(Date.get(position));**

**contentT.setText(Content.get(position));**

**star.setRating(k);**

**}**

**//Set layout**

**return v;**

**}**

**//Create permission of fragment option menu**

**@Override**

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

**super.onCreate(savedInstanceState);**

**setHasOptionsMenu(true);**

**}**

**//Create fragment2 option menu**

**@Override**

**public void onCreateOptionsMenu(Menu menu, MenuInflater inflater) {**

**super.onCreateOptionsMenu(menu, inflater);**

**inflater.inflate( R.menu.fragmentmenu, menu);**

**}**

**//Event listener of fragment2 option menu**

**@Override**

**public boolean onOptionsItemSelected(MenuItem item) {**

**// Handle item selection**

**switch (item.getItemId()) { //Check id**

**case R.id.Back: //If user click 'Back' item**

**//Create fragment manager**

**FragmentManager fragmentManager = getFragmentManager();**

**FragmentTransaction fragmentTransaction = fragmentManager.beginTransaction();**

**//get each fragments id**

**Fragment fragment1 = getActivity().getFragmentManager().findFragmentById(R.id.Fragment1);**

**Fragment fragment2 = getActivity().getFragmentManager().findFragmentById(R.id.Fragment2);**

**//if it is not landscape mode**

**if(fragment1.isHidden()){ //it show list fragment and hide this fragment**

**fragmentTransaction.hide(fragment2);**

**fragmentTransaction.show(fragment1);**

**fragmentTransaction.commit(); //commit transaction**

**}**

**return true;**

**case R.id.Modify: //if click 'Modify', start Modify activity**

**startActivity(new Intent("com.example.homework4.Modify"));**

**getActivity().finish(); //this activity which has this fragment close**

**return true;**

**case R.id.Delete: //if click 'Delete'**

**//get position from preference**

**int position = sh\_pr.getInt("position",0);**

**//Using get data from now task**

**ArrayList<String> Title = new ArrayList<String>();**

**ArrayList<String> Date = new ArrayList<String>();**

**ArrayList<String> Level = new ArrayList<String>();**

**ArrayList<String> Content = new ArrayList<String>();**

**//XML parsing**

**try{**

**//open XML file**

**InputStream In = getActivity().openFileInput(File);**

**if(In != null){**

**XmlPullParserFactory xf = XmlPullParserFactory.newInstance();**

**XmlPullParser parser = xf.newPullParser();**

**// XmlPullParser에 XML 데이터와 인코딩 방식을 입력**

**parser.setInput(In, "utf-8");**

**// czc.xml 을 가져와 XmlPullParser 에 넣는다.**

**// 파싱한 xml 이 END\_DOCUMENT(종료 태그)가 나올때 까지 반복한다.**

**while(parser.getEventType()!=XmlPullParser.END\_DOCUMENT){**

**// 태그의 첫번째 속성일 때,**

**if(parser.getEventType()==XmlPullParser.START\_TAG){**

**// 이름이 "task" 일때, 첫번째 속성값을 ArrayList에 저장**

**if(parser.getName().equals("title")){**

**Title.add(parser.getAttributeValue(0));**

**}**

**if(parser.getName().equals("date")){**

**Date.add(parser.getAttributeValue(0));**

**}**

**if(parser.getName().equals("level")){**

**Level.add(parser.getAttributeValue(0));**

**}**

**if(parser.getName().equals("content")){**

**Content.add(parser.getAttributeValue(0));**

**}**

**}**

**// 다음 태그로 이동**

**parser.next();**

**}**

**}**

**}catch (IllegalArgumentException e) {**

**// TODO Auto-generated catch block**

**e.printStackTrace();**

**} catch (IllegalStateException e) {**

**// TODO Auto-generated catch block**

**e.printStackTrace();**

**} catch (IOException e) {**

**// TODO Auto-generated catch block**

**e.printStackTrace();**

**} catch (XmlPullParserException e) {**

**// TODO Auto-generated catch block**

**e.printStackTrace();**

**}**

**//Remove now task**

**Title.remove(position);**

**Date.remove(position);**

**Level.remove(position);**

**Content.remove(position);**

**String str2="";**

**//and Store now list (remove now task)**

**try {**

**//Open file, Create writer**

**FileOutputStream fOut = getActivity().openFileOutput(File,Context.MODE\_PRIVATE);**

**OutputStreamWriter osw= new OutputStreamWriter(fOut);**

**//Store task using XML shape**

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

**str2 = str2 + "\n" + "<title value = \"" + Title.get(i) + "\">" +"</title>";**

**str2 = str2 + "\n" + "<date value = \"" + Date.get(i) + "\">" + "</date>";**

**str2 = str2 + "\n" + "<content value = \"" + Content.get(i) + "\">" + "</content>";**

**str2 = str2 + "\n" + "<level value = \"" + Level.get(i) + "\">" + "</level>";**

**}**

**osw.write(str2); //write string**

**osw.close(); //close file**

**}catch (IllegalArgumentException e) {**

**// TODO Auto-generated catch block**

**e.printStackTrace();**

**}catch (IllegalStateException e) {**

**// TODO Auto-generated catch block**

**e.printStackTrace();**

**} catch (IOException e) {**

**// TODO Auto-generated catch block**

**e.printStackTrace();**

**}**

**//close this activity and reset**

**startActivity(new Intent("com.example.homework4.loginActivity"));**

**getActivity().finish();**

**return true;**

**default:**

**return super.onOptionsItemSelected(item);**

**}**

**}**

**}**

**<Show Modify Activity = ‘Modify’>**

**package com.example.homework4;**

**import java.io.BufferedOutputStream;**

**import java.io.FileOutputStream;**

**import java.io.IOException;**

**import java.io.InputStream;**

**import java.io.OutputStreamWriter;**

**import java.util.ArrayList;**

**import org.xmlpull.v1.XmlPullParser;**

**import org.xmlpull.v1.XmlPullParserException;**

**import org.xmlpull.v1.XmlPullParserFactory;**

**import android.app.Activity;**

**import android.content.Intent;**

**import android.content.SharedPreferences;**

**import android.os.Bundle;**

**import android.view.View;**

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

**import android.widget.Button;**

**import android.widget.EditText;**

**import android.widget.RatingBar;**

**import android.widget.RatingBar.OnRatingBarChangeListener;**

**//Modify selected task**

**public class Modify extends Activity implements OnClickListener{**

**EditText titleT, dateT, contentT;**

**SharedPreferences k;**

**Button okay;**

**RatingBar Star;**

**float sizeStar=0;**

**BufferedOutputStream bos;**

**String File = "czc.xml";**

**ArrayList<String> Title, Date, Level, Content ;**

**int position;**

**public static Activity mainActivity;**

**@Override**

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

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

**super.onCreate(savedInstanceState);**

**setContentView(R.layout.modify);**

**//using reset activity**

**mainActivity = this;**

**//set each view id to java**

**titleT = (EditText)findViewById(R.id.titleTextModify);**

**dateT = (EditText)findViewById(R.id.dueDateTextModify);**

**contentT = (EditText)findViewById(R.id.contentModify);**

**Star = (RatingBar)findViewById(R.id.ratingBar1Modify);**

**Star.setOnRatingBarChangeListener(new OnRatingBarChangeListener() {**

**//get size of rating bar**

**@Override**

**public void onRatingChanged(RatingBar ratingBar, float rating,**

**boolean fromUser) {**

**sizeStar = rating;**

**}**

**});**

**okay = (Button)findViewById(R.id.okModify);**

**//register click listener**

**okay.setOnClickListener(this);**

**//Using set data to fragment2**

**Title = new ArrayList<String>();**

**Date = new ArrayList<String>();**

**Level = new ArrayList<String>();**

**Content = new ArrayList<String>();**

**//XML parsing**

**try{**

**//open file**

**InputStream In = openFileInput(File);**

**if(In != null){**

**XmlPullParserFactory xf = XmlPullParserFactory.newInstance();**

**XmlPullParser parser = xf.newPullParser();**

**// XmlPullParser에 XML 데이터와 인코딩 방식을 입력**

**parser.setInput(In, "utf-8");**

**// czc.xml 을 가져와 XmlPullParser 에 넣는다.**

**// 파싱한 xml 이 END\_DOCUMENT(종료 태그)가 나올때 까지 반복한다.**

**while(parser.getEventType()!=XmlPullParser.END\_DOCUMENT){**

**// 태그의 첫번째 속성일 때,**

**if(parser.getEventType()==XmlPullParser.START\_TAG){**

**// 이름이 "title" 일때, 첫번째 속성값을 ArrayList에 저장**

**if(parser.getName().equals("title")){**

**Title.add(parser.getAttributeValue(0));**

**}//이름이 "date" 일때, 첫번째 속성값을 ArrayList에 저장**

**if(parser.getName().equals("date")){**

**Date.add(parser.getAttributeValue(0));**

**}//이름이 "level" 일때, 첫번째 속성값을 ArrayList에 저장**

**if(parser.getName().equals("level")){**

**Level.add(parser.getAttributeValue(0));**

**}//이름이 "content" 일때, 첫번째 속성값을 ArrayList에 저장**

**if(parser.getName().equals("content")){**

**Content.add(parser.getAttributeValue(0));**

**}**

**}**

**// 다음 태그로 이동**

**parser.next();**

**}**

**}**

**}catch (IllegalArgumentException e) {**

**// TODO Auto-generated catch block**

**e.printStackTrace();**

**} catch (IllegalStateException e) {**

**// TODO Auto-generated catch block**

**e.printStackTrace();**

**} catch (IOException e) {**

**// TODO Auto-generated catch block**

**e.printStackTrace();**

**} catch (XmlPullParserException e) {**

**// TODO Auto-generated catch block**

**e.printStackTrace();**

**}**

**//get preference**

**k = getSharedPreferences("My ToDo-list",0);**

**//get position this task**

**position = k.getInt("position", 0);**

**//set views for initialize first display**

**float size = Float.parseFloat(Level.get(position));**

**titleT.setText(Title.get(position));**

**dateT.setText(Date.get(position));**

**contentT.setText(Content.get(position));**

**Star.setRating(size);**

**}**

**//Button click listener**

**@Override**

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

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

**//if user click modify okay button**

**if(v.getId() == okay.getId()){**

**//float convert to string for store XML**

**String sizeRating = ""+sizeStar;**

**//Restore modify task to each array list**

**Title.set(position, titleT.getText().toString());**

**Date.set(position,dateT.getText().toString());**

**Level.set(position, sizeRating);**

**Content.set(position, contentT.getText().toString());**

**//Restore to XML file**

**String str2="";**

**try {**

**//open XML file and create writer**

**FileOutputStream fOut = openFileOutput(File,MODE\_PRIVATE);**

**OutputStreamWriter osw= new OutputStreamWriter(fOut);**

**//Input data using XML shape**

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

**str2 = str2 + "\n" + "<title value = \"" + Title.get(i) + "\">" +"</title>";**

**str2 = str2 + "\n" + "<date value = \"" + Date.get(i) + "\">" + "</date>";**

**str2 = str2 + "\n" + "<content value = \"" + Content.get(i) + "\">" + "</content>";**

**str2 = str2 + "\n" + "<level value = \"" + Level.get(i) + "\">" + "</level>";**

**}**

**osw.write(str2); //Write data**

**osw.close(); //close file**

**}catch (IllegalArgumentException e) {**

**// TODO Auto-generated catch block**

**e.printStackTrace();**

**}catch (IllegalStateException e) {**

**// TODO Auto-generated catch block**

**e.printStackTrace();**

**} catch (IOException e) {**

**// TODO Auto-generated catch block**

**e.printStackTrace();**

**}**

**//reset main list Activity, close this activity**

**mainActivity.finish();**

**startActivity(new Intent("com.example.homework4.loginActivity"));**

**}**

**}**

**//It is used to reset task list And concern user click back button**

**@Override**

**public void onBackPressed(){**

**//Execute main board activity (it has 2 fragment)**

**startActivity(new Intent("com.example.homework4.loginActivity"));**

**finish();**

**}**

**}**

**<Show Add task Activity = ‘AddWork’>**

**package com.example.homework4;**

**import java.io.BufferedReader;**

**import java.io.FileOutputStream;**

**import java.io.IOException;**

**import java.io.InputStream;**

**import java.io.InputStreamReader;**

**import java.io.OutputStreamWriter;**

**import java.text.SimpleDateFormat;**

**import android.app.Activity;**

**import android.content.Intent;**

**import android.os.Bundle;**

**import android.view.View;**

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

**import android.widget.Button;**

**import android.widget.EditText;**

**import android.widget.RatingBar;**

**import android.widget.RatingBar.OnRatingBarChangeListener;**

**//This activity add user task**

**public class AddWork extends Activity implements OnClickListener{**

**Button okay; //It is used to Add task to XML file**

**EditText Title, Date, Content; //Set text to views**

**RatingBar Star; //Set rating (importance of task)**

**float sizeStar=0; //initialize**

**String File = "czc.xml"; //File name**

**String register; //It register value to XML which has task information**

**String str,str2=" "; //it is used to initialize arraylist**

**//get current activity information**

**public static Activity mainActivity;**

**@Override**

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

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

**super.onCreate(savedInstanceState);**

**setContentView(R.layout.addwork);**

**mainActivity = this; //set current activity information**

**//Set id from xml view**

**Title = (EditText)findViewById(R.id.titleText);**

**Date = (EditText)findViewById(R.id.dueDateText);**

**Content = (EditText)findViewById(R.id.content);**

**Star = (RatingBar)findViewById(R.id.ratingBar1);**

**//get size of level information from rating bar**

**Star.setOnRatingBarChangeListener(new OnRatingBarChangeListener() {**

**@Override**

**public void onRatingChanged(RatingBar ratingBar, float rating,**

**boolean fromUser) {**

**sizeStar = rating; //get importance of task**

**}**

**});**

**//set button click listener**

**okay = (Button)findViewById(R.id.ok);**

**okay.setOnClickListener(this);**

**}**

**@Override**

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

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

**//if user click button okay**

**if(v.getId() == okay.getId()){**

**//Convert float to string which store to XML file**

**String sizeRating = ""+sizeStar; //Float convert to String**

**String S\_title,S\_date,S\_content; //get task information from user**

**S\_title = Title.getText().toString();**

**S\_date = Date.getText().toString();**

**S\_content = Content.getText().toString();**

**try {**

**//Open file**

**InputStream in = openFileInput(File);**

**if (in!=null) {**

**//Set buffer reader**

**BufferedReader reader = new BufferedReader(new InputStreamReader(in));**

**str = "";**

**StringBuffer buf = new StringBuffer();**

**while ((str = reader.readLine()) != null) {**

**//get all task inforatmion**

**buf.append(str+"\n");**

**}**

**if(str2!=null)**

**str2 = buf.toString(); //set string from buffer**

**in.close();**

**}**

**}catch (java.io.FileNotFoundException e) {**

**}**

**catch (Throwable t2) {**

**}**

**try {**

**//open file to write**

**FileOutputStream fOut = openFileOutput(File,MODE\_PRIVATE);**

**//create writer**

**OutputStreamWriter osw= new OutputStreamWriter(fOut);**

**//Set data using XML shape**

**register = "<title value = \"" + S\_title + "\">" +"</title>";**

**register = register + "\n" + "<date value = \"" + S\_date + "\">" + "</date>";**

**register = register + "\n" + "<content value = \"" + S\_content + "\">" + "</content>";**

**register = register + "\n" + "<level value = \"" + sizeRating + "\">" + "</level>";**

**//Restore all task (add preview data and current task)**

**osw.write(str2+"\n"+register);**

**osw.close();**

**}catch (IllegalArgumentException e) {**

**// TODO Auto-generated catch block**

**e.printStackTrace();**

**}catch (IllegalStateException e) {**

**// TODO Auto-generated catch block**

**e.printStackTrace();**

**} catch (IOException e) {**

**// TODO Auto-generated catch block**

**e.printStackTrace();**

**}**

**//Reset list fragment and close this activity**

**mainActivity.finish();**

**startActivity(new Intent("com.example.homework4.loginActivity"));**

**}**

**}**

**//It is used to reset task list And concern user click back button**

**@Override**

**public void onBackPressed(){**

**//Execute main board activity (it has 2 fragment)**

**startActivity(new Intent("com.example.homework4.loginActivity"));**

**finish();**

**}**

**}**

**<My task information using sort task by title = ‘MyTask’>**

**package** com.example.homework4;

//It has task information

//and it is used to sort task

**public** **class** MyTask {

String title;

String date;

String level;

String content;

**public** MyTask(){ //Set default constructor

title = **null**;

date = **null**;

level = **null**;

content = **null**;

}

//Set task information

**public** **void** setMyTask(String a, String b, String c, String d){

title = a;

date = b;

level = c;

content = d;

}

//Initialize date

**public** **void** setDate(String a){

date = a;

}

//get information of each data

**public** String getTitle(){

**return** title;

}

**public** String getDDate(){

**return** date;

}

**public** String getLevel(){

**return** level;

}

**public** String getContent(){

**return** content;

}

}