package com.nhc.session2;  
  
import androidx.annotation.Nullable;  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.app.DatePickerDialog;  
import android.content.Intent;  
import android.os.Bundle;  
import android.view.View;  
import android.widget.AdapterView;  
import android.widget.ArrayAdapter;  
import android.widget.Button;  
import android.widget.DatePicker;  
import android.widget.EditText;  
import android.widget.ListView;  
import android.widget.Spinner;  
import android.widget.TextView;  
  
import com.google.android.material.floatingactionbutton.FloatingActionButton;  
import com.nhc.session2.Class.AssetGroups;  
import com.nhc.session2.Class.Assets;  
import com.nhc.session2.Class.Departments;  
import com.nhc.session2.Adapter.AssetListAdapter;  
  
import java.sql.SQLException;  
import java.util.ArrayList;  
import java.util.Calendar;  
  
public class AssetActivity extends AppCompatActivity {  
  
 EditText e\_startdate, e\_endate, e\_search;  
 Spinner sp\_department, sp\_assetgroup;  
 Button btn\_search, btn\_refresh;  
 static ListView *lv\_asset*;  
 static TextView *tv\_amount*;  
 FloatingActionButton fab;  
  
 static ArrayList<Assets> *assets* = new ArrayList<>();  
 static long *allAmount*;  
 private int requestCode = 123;  
  
 private int lastSelectedYearS;  
 private int lastSelectedMonthS;  
 private int lastSelectedDayOfMonthS;  
 private int lastSelectedYearE;  
 private int lastSelectedMonthE;  
 private int lastSelectedDayOfMonthE;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_asset*);  
  
 e\_startdate = findViewById(R.id.*e\_startdate*);  
 e\_endate = findViewById(R.id.*e\_enddate*);  
 e\_search = findViewById(R.id.*e\_search*);  
 sp\_department = findViewById(R.id.*sp\_department*);  
 sp\_assetgroup = findViewById(R.id.*sp\_assetgroup*);  
 btn\_search = findViewById(R.id.*btn\_search*);  
 btn\_refresh = findViewById(R.id.*btn\_refresh*);  
 *lv\_asset* = findViewById(R.id.*lv\_asset*);  
 *tv\_amount* = findViewById(R.id.*tv\_amount*);  
 fab = findViewById(R.id.*fab*);  
  
 e\_startdate.setFocusable(false);  
 e\_startdate.setCursorVisible(false);  
 e\_endate.setFocusable(false);  
 e\_endate.setCursorVisible(false);  
  
 final Calendar c = Calendar.*getInstance*();  
 this.lastSelectedYearS = c.get(Calendar.*YEAR*);  
 this.lastSelectedMonthS = c.get(Calendar.*MONTH*);  
 this.lastSelectedDayOfMonthS = c.get(Calendar.*DAY\_OF\_MONTH*);  
  
 this.lastSelectedYearE = this.lastSelectedYearS;  
 this.lastSelectedMonthE = this.lastSelectedMonthS;  
 this.lastSelectedDayOfMonthE = this.lastSelectedDayOfMonthS;  
  
 fab.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 Intent intent = new Intent(AssetActivity.this, AssetInfomationActivity.class);  
 intent.putExtra("status", "add");  
 startActivityForResult(intent, requestCode);  
 }  
 });  
  
 e\_startdate.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 SelectDateS();  
 }  
 });  
 e\_startdate.setOnLongClickListener(new View.OnLongClickListener() {  
 @Override  
 public boolean onLongClick(View v) {  
 e\_startdate.setText("");  
 try {  
 refresh(caseValue());  
 } catch (SQLException throwables) {  
 throwables.printStackTrace();  
 }  
 return false;  
 }  
 });  
  
 e\_endate.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 SelectDateE();  
 }  
 });  
 e\_endate.setOnLongClickListener(new View.OnLongClickListener() {  
 @Override  
 public boolean onLongClick(View v) {  
 e\_endate.setText("");  
 try {  
 refresh(caseValue());  
 } catch (SQLException throwables) {  
 throwables.printStackTrace();  
 }  
 return false;  
 }  
 });  
  
 ArrayList<String> department = new ArrayList<>();  
 ArrayList<String> assetGroup = new ArrayList<>();  
 department.add("- Department -");  
 assetGroup.add("- Asset Group -");  
 try {  
 ArrayList<Departments> departments = new Session2Model().getDepartmentList();  
 ArrayList<AssetGroups> assetGroups = new Session2Model().getAssetGroupsList();  
  
 for (Departments d : departments){  
 department.add(d.getName());  
 }  
 for(AssetGroups a : assetGroups){  
 assetGroup.add(a.getName());  
 }  
  
 ArrayAdapter<String> departmentsArrayAdapter = new ArrayAdapter<String>(this, android.R.layout.*simple\_spinner\_item*, department);  
 departmentsArrayAdapter.setDropDownViewResource(R.layout.*support\_simple\_spinner\_dropdown\_item*);  
 sp\_department.setAdapter(departmentsArrayAdapter);  
 ArrayAdapter<String> assetGroupsArrayAdapter = new ArrayAdapter<String>(this, android.R.layout.*simple\_spinner\_item*, assetGroup);  
 assetGroupsArrayAdapter.setDropDownViewResource(R.layout.*support\_simple\_spinner\_dropdown\_item*);  
 sp\_assetgroup.setAdapter(assetGroupsArrayAdapter);  
  
 } catch (SQLException throwables) {  
 throwables.printStackTrace();  
 }  
  
 try {  
 load();  
 } catch (SQLException throwables) {  
 throwables.printStackTrace();  
 }  
  
 btn\_search.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 try {  
 search();  
 } catch (SQLException throwables) {  
 throwables.printStackTrace();  
 }  
 }  
 });  
  
 btn\_refresh.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 try {  
 load();  
 } catch (SQLException throwables) {  
 throwables.printStackTrace();  
 }  
 }  
 });  
  
 sp\_department.setOnItemSelectedListener(new AdapterView.OnItemSelectedListener() {  
 @Override  
 public void onItemSelected(AdapterView<?> parent, View view, int position, long id) {  
 try {  
 refresh(caseValue());  
 } catch (SQLException throwables) {  
 throwables.printStackTrace();  
 }  
 }  
  
 @Override  
 public void onNothingSelected(AdapterView<?> parent) {  
  
 }  
 });  
  
 sp\_assetgroup.setOnItemSelectedListener(new AdapterView.OnItemSelectedListener() {  
 @Override  
 public void onItemSelected(AdapterView<?> parent, View view, int position, long id) {  
 try {  
 refresh(caseValue());  
 } catch (SQLException throwables) {  
 throwables.printStackTrace();  
 }  
 }  
  
 @Override  
 public void onNothingSelected(AdapterView<?> parent) {  
  
 }  
 });  
  
 }  
  
 @Override  
 protected void onActivityResult(int requestCode, int resultCode, @Nullable Intent data) {  
 super.onActivityResult(requestCode, resultCode, data);  
 try {  
 load();  
 } catch (SQLException throwables) {  
 throwables.printStackTrace();  
 }  
 }  
  
 private int caseValue(){  
 boolean startDateSelected = !e\_startdate.getText().toString().isEmpty();  
 boolean endDateSelected = !e\_endate.getText().toString().isEmpty();  
 if(sp\_department.getSelectedItemId() == 0 && sp\_assetgroup.getSelectedItemId() == 0){  
 if(!startDateSelected && !endDateSelected){  
 *// no select* return 0;  
 }  
 else if(startDateSelected && !endDateSelected){  
 *// S* return 3;  
 }  
 else if (!startDateSelected && endDateSelected){  
 *// E* return 4;  
 }  
 else {  
 *// SE* return 10;  
 }  
 }  
 else if(sp\_department.getSelectedItemId() == 0 && sp\_assetgroup.getSelectedItemId() != 0){  
 if(!startDateSelected && !endDateSelected){  
 *// A* return 2;  
 }  
 else if(startDateSelected && !endDateSelected){  
 *// AS* return 8;  
 }  
 else if (!startDateSelected && endDateSelected){  
 *// AE* return 9;  
 }  
 else {  
 *// ASE* return 14;  
 }  
 }  
 else if(sp\_department.getSelectedItemId() != 0 && sp\_assetgroup.getSelectedItemId() == 0){  
 if(!startDateSelected && !endDateSelected){  
 *// D* return 1;  
 }  
 else if(startDateSelected && !endDateSelected){  
 *// DS* return 6;  
 }  
 else if (!startDateSelected && endDateSelected){  
 *// DE* return 7;  
 }  
 else {  
 *// DSE* return 13;  
 }  
 }  
 else {  
 if(!startDateSelected && !endDateSelected){  
 *// DA* return 5;  
 }  
 else if(startDateSelected && !endDateSelected){  
 *// DAS* return 11;  
 }  
 else if (!startDateSelected && endDateSelected){  
 *// DAE* return 12;  
 }  
 else {  
 *// DASE* return 15;  
 }  
 }  
 }  
  
 private void SelectDateS(){  
 DatePickerDialog.OnDateSetListener dateSetListener = new DatePickerDialog.OnDateSetListener() {  
 @Override  
 public void onDateSet(DatePicker view, int year, int month, int dayOfMonth) {  
 e\_startdate.setText(year + "-" + (month + 1) + "-" + dayOfMonth);  
 lastSelectedYearS = year;  
 lastSelectedMonthS = month+1;  
 lastSelectedDayOfMonthS = dayOfMonth;  
 try {  
 refresh(caseValue());  
 } catch (SQLException throwables) {  
 throwables.printStackTrace();  
 }  
 }  
 };  
 DatePickerDialog datePickerDialog = null;  
 datePickerDialog = new DatePickerDialog(this, android.R.style.*Theme\_Holo\_Light\_Dialog\_NoActionBar*, dateSetListener, lastSelectedYearS, lastSelectedMonthS, lastSelectedDayOfMonthS);  
 datePickerDialog.show();  
 }  
  
 private void SelectDateE(){  
 DatePickerDialog.OnDateSetListener dateSetListener = new DatePickerDialog.OnDateSetListener() {  
 @Override  
 public void onDateSet(DatePicker view, int year, int month, int dayOfMonth) {  
 e\_endate.setText(year + "-" + (month + 1) + "-" + dayOfMonth);  
 lastSelectedYearE = year;  
 lastSelectedMonthE = month+1;  
 lastSelectedDayOfMonthE = dayOfMonth;  
 try {  
 refresh(caseValue());  
 } catch (SQLException throwables) {  
 throwables.printStackTrace();  
 }  
 }  
 };  
 DatePickerDialog datePickerDialog = null;  
 datePickerDialog = new DatePickerDialog(this, android.R.style.*Theme\_Holo\_Light\_Dialog\_NoActionBar*, dateSetListener, lastSelectedYearE, lastSelectedMonthE, lastSelectedDayOfMonthE);  
 datePickerDialog.show();  
 }  
  
 private void load() throws SQLException {  
 *assets* = new Session2Model().getAssetsList();  
 *allAmount* = *assets*.size();  
 AssetListAdapter assetListAdapter = new AssetListAdapter(this, *assets*);  
 *lv\_asset*.setAdapter(assetListAdapter);  
 String s\_amount = *allAmount* + " assets from " + *allAmount*;  
 *tv\_amount*.setText(s\_amount);  
 sp\_assetgroup.setSelection(0);  
 sp\_department.setSelection(0);  
 e\_startdate.setText("");  
 e\_endate.setText("");  
 e\_search.setText("");  
 }  
  
 static public void loadAllAssetAmount(long amount){  
 *allAmount* = amount;  
 String s\_amount = *assets*.size() + " assets from " + *allAmount*;  
 *tv\_amount*.setText(s\_amount);  
 }  
  
 private void search() throws SQLException {  
 String search = e\_search.getText().toString();  
 if(!search.isEmpty()){  
 ArrayList<Assets> t = new ArrayList<>();  
 for (Assets a : *assets*){  
 if(a.getAssetName().contains(search) || a.getAssetSN().contains(search)) t.add(a);  
 }  
 String s\_amount = t.size() + " assets from " + *allAmount*;  
  
 AssetListAdapter assetListAdapter = new AssetListAdapter(this, t);  
 *lv\_asset*.setAdapter(assetListAdapter);  
 *tv\_amount*.setText(s\_amount);  
 }  
 else refresh(caseValue());  
 }  
  
 private void refresh(int cases) throws SQLException{  
 switch (cases){  
 case 1:  
 *assets* = new Session2Model().getAssetsListByD(sp\_department.getSelectedItemId());  
 break;  
 case 2:  
 *assets* = new Session2Model().getAssetsListByA(sp\_assetgroup.getSelectedItemId());  
 break;  
 case 3:  
 *assets* = new Session2Model().getAssetsListByS(e\_startdate.getText().toString().trim());  
 break;  
 case 4:  
 *assets* = new Session2Model().getAssetsListByE(e\_endate.getText().toString().trim());  
 break;  
 case 5:  
 *assets* = new Session2Model().getAssetsListByDA(sp\_department.getSelectedItemId(), sp\_assetgroup.getSelectedItemId());  
 break;  
 case 6:  
 *assets* = new Session2Model().getAssetsListByDS(sp\_department.getSelectedItemId(), e\_startdate.getText().toString().trim());  
 break;  
 case 7:  
 *assets* = new Session2Model().getAssetsListByDE(sp\_department.getSelectedItemId(), e\_endate.getText().toString().trim());  
 break;  
 case 8:  
 *assets* = new Session2Model().getAssetsListByAS(sp\_assetgroup.getSelectedItemId(), e\_startdate.getText().toString().trim());  
 break;  
 case 9:  
 *assets* = new Session2Model().getAssetsListByAE(sp\_assetgroup.getSelectedItemId(), e\_endate.getText().toString().trim());  
 break;  
 case 10:  
 *assets* = new Session2Model().getAssetsListBySE(e\_startdate.getText().toString().trim(), e\_endate.getText().toString().trim());  
 break;  
 case 11:  
 *assets* = new Session2Model().getAssetsListByDAS(sp\_department.getSelectedItemId(), sp\_assetgroup.getSelectedItemId(), e\_startdate.getText().toString().trim());  
 break;  
 case 12:  
 *assets* = new Session2Model().getAssetsListByDAE(sp\_department.getSelectedItemId(), sp\_assetgroup.getSelectedItemId(), e\_endate.getText().toString().trim());  
 break;  
 case 13:  
 *assets* = new Session2Model().getAssetsListByDSE(sp\_department.getSelectedItemId(), e\_startdate.getText().toString().trim(), e\_endate.getText().toString().trim());  
 break;  
 case 14:  
 *assets* = new Session2Model().getAssetsListByASE(sp\_assetgroup.getSelectedItemId(), e\_startdate.getText().toString().trim(), e\_endate.getText().toString().trim());  
 break;  
 case 15:  
 *assets* = new Session2Model().getAssetsListByDASE(sp\_department.getSelectedItemId(), sp\_assetgroup.getSelectedItemId(), e\_startdate.getText().toString().trim(), e\_endate.getText().toString().trim());  
 break;  
 default:  
 load();  
 }  
 String s\_amount = *assets*.size() + " assets from " + *allAmount*;  
*// if(!e\_search.getText().toString().isEmpty()){  
// String s = e\_search.getText().toString();  
// ArrayList<Assets> t = new ArrayList<>();  
// for (Assets a : assets){  
// if(a.getAssetName().contains(s)) t.add(a);  
// }  
// assets.clear();  
// assets = t;  
// s\_amount = t.size() + " assets from " + allAmount;  
// }* AssetListAdapter assetListAdapter = new AssetListAdapter(this, *assets*);  
 *lv\_asset*.setAdapter(assetListAdapter);  
 *tv\_amount*.setText(s\_amount);  
 }  
}