package com.nhc.session2;  
  
import com.nhc.session2.Class.AssetGroups;  
import com.nhc.session2.Class.Assets;  
import com.nhc.session2.Class.DepartmentLocations;  
import com.nhc.session2.Class.Departments;  
import com.nhc.session2.Class.Employees;  
import com.nhc.session2.Class.Locations;  
import com.nhc.session2.JDBC.Controllers.JDBCController;  
  
import java.sql.Connection;  
import java.sql.PreparedStatement;  
import java.sql.ResultSet;  
import java.sql.SQLException;  
import java.sql.Statement;  
import java.util.ArrayList;  
  
public class Session2Model {  
 private JDBCController jdbcController = new JDBCController();  
 private Connection connection;  
  
 public Session2Model(){  
 connection = jdbcController.ConnnectionData();  
 }  
  
 public ArrayList<Employees> getEmployeesList() throws SQLException {  
 ArrayList<Employees> list = new ArrayList<>();  
 Statement statement = connection.createStatement();  
 String sql = "select \* from Employees";  
 ResultSet rs = statement.executeQuery(sql);  
 while (rs.next()) {  
 list.add(new Employees(rs.getLong("ID"), rs.getString("FirstName"), rs.getString("LastName"),  
 rs.getString("Phone"), rs.getBoolean("isAdmin"), rs.getString("Username"), rs.getString("Password")));  
 }  
 connection.close();  
 return list;  
 }  
  
 public ArrayList<Departments> getDepartmentList() throws SQLException {  
 ArrayList<Departments> list = new ArrayList<>();  
 Statement statement = connection.createStatement();  
 String sql = "select \* from Departments";  
 ResultSet rs = statement.executeQuery(sql);  
 while (rs.next()) {  
 list.add(new Departments(rs.getLong("ID"), rs.getString("Name")));  
 }  
 connection.close();  
 return list;  
 }  
  
 public ArrayList<AssetGroups> getAssetGroupsList() throws SQLException {  
 ArrayList<AssetGroups> list = new ArrayList<>();  
 Statement statement = connection.createStatement();  
 String sql = "select \* from AssetGroups";  
 ResultSet rs = statement.executeQuery(sql);  
 while (rs.next()) {  
 list.add(new AssetGroups(rs.getLong("ID"), rs.getString("Name")));  
 }  
 connection.close();  
 return list;  
 }  
  
 public ArrayList<Locations> getLocationsList() throws SQLException {  
 ArrayList<Locations> list = new ArrayList<>();  
 Statement statement = connection.createStatement();  
 String sql = "select \* from Locations";  
 ResultSet rs = statement.executeQuery(sql);  
 while (rs.next()) {  
 list.add(new Locations(rs.getLong("ID"), rs.getString("Name")));  
 }  
 connection.close();  
 return list;  
 }  
  
 public ArrayList<DepartmentLocations> getDepartmentLocationsList() throws SQLException {  
 ArrayList<DepartmentLocations> list = new ArrayList<>();  
 Statement statement = connection.createStatement();  
 String sql = "select \* from DepartmentLocations";  
 ResultSet rs = statement.executeQuery(sql);  
 while (rs.next()) {  
 list.add(new DepartmentLocations(rs.getLong("ID"), rs.getLong("DepartmentID"),  
 rs.getLong("LocationID"), rs.getString("StartDate"), rs.getString("EndDate")));  
 }  
 connection.close();  
 return list;  
 }  
  
 public ArrayList<Assets> getAssetsList() throws SQLException {  
 ArrayList<Assets> list = new ArrayList<>();  
 Statement statement = connection.createStatement();  
 String sql = "select \* from Assets";  
 ResultSet rs = statement.executeQuery(sql);  
 while (rs.next()) {  
 list.add(new Assets(rs.getLong("ID"), rs.getString("AssetSN"), rs.getString("AssetName"), rs.getLong("DepartmentLocationID"),  
 rs.getLong("EmployeeID"), rs.getLong("AssetGroupID"), rs.getString("Description"), rs.getString("WarrantyDate")));  
 }  
 connection.close();  
 return list;  
 }  
  
 public String getDepartmentName(long departmentID) throws SQLException {  
 String name = "";  
 Statement statement = connection.createStatement();  
 String sql = "select \* from Departments where ID = " + departmentID;  
 ResultSet rs = statement.executeQuery(sql);  
 while (rs.next()) {  
 name = rs.getString("Name");  
 }  
 connection.close();  
 return name;  
 }  
  
 public long getDepartmentID(long departmentLocationID) throws SQLException {  
 long departmentID = -1;  
 Statement statement = connection.createStatement();  
 String sql = "select \* from DepartmentLocations where ID = " + departmentLocationID;  
 ResultSet rs = statement.executeQuery(sql);  
 while (rs.next()) {  
 departmentID = rs.getLong("DepartmentID");  
 }  
 connection.close();  
 return departmentID;  
 }  
  
 public long getDepartmentID(String name) throws SQLException {  
 long ID = -1;  
 Statement statement = connection.createStatement();  
 String sql = "select \* from Departments where Name = '" + name + "'";  
 ResultSet rs = statement.executeQuery(sql);  
 while (rs.next()) {  
 ID = rs.getLong("ID");  
 }  
 connection.close();  
 return ID;  
 }  
  
 public long getLocationID(String name) throws SQLException{  
 long ID = 0;  
 Statement statement = connection.createStatement();  
 String sql = "select \* from Locations where Name = '" + name + "'";  
 ResultSet rs = statement.executeQuery(sql);  
 while (rs.next()) {  
 ID = rs.getLong("ID");  
 }  
 connection.close();  
 return ID;  
 }  
  
 public long getLocationID(long departmentLocationID) throws SQLException{  
 long locationID = -1;  
 Statement statement = connection.createStatement();  
 String sql = "select \* from DepartmentLocations where ID = " + departmentLocationID;  
 ResultSet rs = statement.executeQuery(sql);  
 while (rs.next()) {  
 locationID = rs.getLong("LocationID");  
 }  
 connection.close();  
 return locationID;  
 }  
  
 public long getDepartmentLocationID(long dep\_id, long loc\_id) throws SQLException{  
 long ID = 0;  
 Statement statement = connection.createStatement();  
 String sql = "select ID from DepartmentLocations where DepartmentID = " + dep\_id + " and LocationID = " + loc\_id;  
 ResultSet rs = statement.executeQuery(sql);  
 while (rs.next()) {  
 ID = rs.getLong("ID");  
 }  
 connection.close();  
 return ID;  
 }  
  
 public long getAssetGroupID(String name) throws SQLException {  
 long ID = -1;  
 Statement statement = connection.createStatement();  
 String sql = "select \* from AssetGroups where Name = '" + name + "'";  
 ResultSet rs = statement.executeQuery(sql);  
 while (rs.next()) {  
 ID = rs.getLong("ID");  
 }  
 connection.close();  
 return ID;  
 }  
  
 public ArrayList<Assets> getAssetsList(String sql) throws SQLException {  
 ArrayList<Assets> list = new ArrayList<>();  
 Statement statement = connection.createStatement();  
 ResultSet rs = statement.executeQuery(sql);  
 while (rs.next()) {  
 list.add(new Assets(rs.getLong("ID"), rs.getString("AssetSN"), rs.getString("AssetName"), rs.getLong("DepartmentLocationID"),  
 rs.getLong("EmployeeID"), rs.getLong("AssetGroupID"), rs.getString("Description"), rs.getString("WarrantyDate")));  
 }  
 connection.close();  
 return list;  
 }  
  
 public ArrayList<Assets> getAssetsListByDASE(long departmentID, long assetGroupID, String startDate, String endDate) throws SQLException {  
 ArrayList<Assets> list = new ArrayList<>();  
 String sql = "select \* from Assets where DepartmentLocationID in (select ID from DepartmentLocations where DepartmentID = '" + departmentID  
 + "' and StartDate >= '" + startDate + "' and EndDate <= '" + endDate + "') and AssetGroupID = '" + assetGroupID + "'";  
 list = getAssetsList(sql);  
 return list;  
 }  
  
 public ArrayList<Assets> getAssetsListByD(long departmentID) throws SQLException {  
 ArrayList<Assets> list = new ArrayList<>();  
 String sql = "select \* from Assets where DepartmentLocationID in (select ID from DepartmentLocations where DepartmentID = " + departmentID + ")";  
 list = getAssetsList(sql);  
 return list;  
 }  
  
 public ArrayList<Assets> getAssetsListByA(long assetGroupID) throws SQLException {  
 ArrayList<Assets> list = new ArrayList<>();  
 String sql = "select \* from Assets where AssetGroupID = " + assetGroupID;  
 list = getAssetsList(sql);  
 return list;  
 }  
  
 public ArrayList<Assets> getAssetsListByS(String startDate) throws SQLException {  
 ArrayList<Assets> list = new ArrayList<>();  
 String sql = "select \* from Assets where DepartmentLocationID in (select ID from DepartmentLocations where StartDate >= '" + startDate + "')";  
 list = getAssetsList(sql);  
 return list;  
 }  
  
 public ArrayList<Assets> getAssetsListByE(String endDate) throws SQLException {  
 ArrayList<Assets> list = new ArrayList<>();  
 String sql = "select \* from Assets where DepartmentLocationID in (select ID from DepartmentLocations where EndDate <= '" + endDate + "')";  
 list = getAssetsList(sql);  
 return list;  
 }  
  
 public ArrayList<Assets> getAssetsListByDA(long departmentID, long assetGroupID) throws SQLException {  
 ArrayList<Assets> list = new ArrayList<>();  
 String sql = "select \* from Assets where DepartmentLocationID in (select ID from DepartmentLocations where DepartmentID = " + departmentID + ") and AssetGroupID = " + assetGroupID;  
 list = getAssetsList(sql);  
 return list;  
 }  
  
 public ArrayList<Assets> getAssetsListByDS(long departmentID, String startDate) throws SQLException {  
 ArrayList<Assets> list = new ArrayList<>();  
 String sql = "select \* from Assets where DepartmentLocationID in (select ID from DepartmentLocations where DepartmentID = " + departmentID + " and StartDate >= '" + startDate + "')";  
 list = getAssetsList(sql);  
 return list;  
 }  
  
 public ArrayList<Assets> getAssetsListByDE(long departmentID, String endDate) throws SQLException {  
 ArrayList<Assets> list = new ArrayList<>();  
 String sql = "select \* from Assets where DepartmentLocationID in (select ID from DepartmentLocations where DepartmentID = " + departmentID + " and EndDate >= '" + endDate + "')";  
 list = getAssetsList(sql);  
 return list;  
 }  
  
 public ArrayList<Assets> getAssetsListByAS(long assetGroupID, String startDate) throws SQLException {  
 ArrayList<Assets> list = new ArrayList<>();  
 String sql = "select \* from Assets where DepartmentLocationID in (select ID from DepartmentLocations where StartDate >= '" + startDate + "') and AssetGroupID = " + assetGroupID;  
 list = getAssetsList(sql);  
 return list;  
 }  
  
 public ArrayList<Assets> getAssetsListByAE(long assetGroupID, String endDate) throws SQLException {  
 ArrayList<Assets> list = new ArrayList<>();  
 String sql = "select \* from Assets where DepartmentLocationID in (select ID from DepartmentLocations where EndDate <= '" + endDate + "') and AssetGroupID = " + assetGroupID;  
 list = getAssetsList(sql);  
 return list;  
 }  
  
 public ArrayList<Assets> getAssetsListBySE(String startDate, String endDate) throws SQLException {  
 ArrayList<Assets> list = new ArrayList<>();  
 String sql = "select \* from Assets where DepartmentLocationID in (select ID from DepartmentLocations where StartDate >= '" + startDate + "' and EndDate <= '" + endDate + "')";  
 list = getAssetsList(sql);  
 return list;  
 }  
  
 public ArrayList<Assets> getAssetsListByDAS(long departmentID, long assetGroupID, String startDate) throws SQLException {  
 ArrayList<Assets> list = new ArrayList<>();  
 String sql = "select \* from Assets where DepartmentLocationID in (select ID from DepartmentLocations where DepartmentID = " + departmentID  
 + " and StartDate >= '" + startDate + "') and AssetGroupID = " + assetGroupID;  
 list = getAssetsList(sql);  
 return list;  
 }  
  
 public ArrayList<Assets> getAssetsListByDAE(long departmentID, long assetGroupID, String endDate) throws SQLException {  
 ArrayList<Assets> list = new ArrayList<>();  
 String sql = "select \* from Assets where DepartmentLocationID in (select ID from DepartmentLocations where DepartmentID = " + departmentID  
 + " and StartDate <= '" + endDate + "') and AssetGroupID = " + assetGroupID;  
 list = getAssetsList(sql);  
 return list;  
 }  
  
 public ArrayList<Assets> getAssetsListByDSE(long departmentID, String startDate, String endDate) throws SQLException {  
 ArrayList<Assets> list = new ArrayList<>();  
 String sql = "select \* from Assets where DepartmentLocationID in (select ID from DepartmentLocations where DepartmentID = '" + departmentID  
 + "' and StartDate >= '" + startDate + "' and EndDate <= '" + endDate + "')";  
 list = getAssetsList(sql);  
 return list;  
 }  
  
 public ArrayList<Assets> getAssetsListByASE(long assetGroupID, String startDate, String endDate) throws SQLException {  
 ArrayList<Assets> list = new ArrayList<>();  
 String sql = "select \* from Assets where DepartmentLocationID in (select ID from DepartmentLocations where StartDate >= '" + startDate  
 + "' and EndDate <= '" + endDate + "') and AssetGroupID = '" + assetGroupID + "'";  
 list = getAssetsList(sql);  
 return list;  
 }  
  
 public long login(String username, String password) throws SQLException{  
 long ID = -1;  
 Statement statement = connection.createStatement();  
 String sql = "select \* from Employees where Username = '" + username + "' and Password = '" + password + "'";  
 ResultSet rs = statement.executeQuery(sql);  
 while (rs.next()) {  
 ID = rs.getLong("ID");  
 }  
 connection.close();  
 return ID;  
 }  
  
 public boolean isAdmin(long ID) throws SQLException{  
 Statement statement = connection.createStatement();  
 String sql = "select \* from Employees where ID = " + ID;  
 ResultSet rs = statement.executeQuery(sql);  
 while (rs.next()) {  
 if (rs.getBoolean("isAdmin")) {  
 connection.close();  
 return true;  
 }  
 }  
 connection.close();  
 return false;  
 }  
  
 public boolean AssetSNisExists(String sn) throws SQLException {  
 Statement statement = connection.createStatement();  
 String sql = "select \* from Assets where AssetSN = '" + sn + "'";  
 ResultSet rs = statement.executeQuery(sql);  
 while (rs.next()) {  
 long id = 0;  
 id = rs.getLong("ID");  
 if (id > 0) {  
 connection.close();  
 return true;  
 }  
 }  
 connection.close();  
 return false;  
 }  
  
 public Employees getEmployees(long id) throws SQLException{  
 ArrayList<Employees> list = new ArrayList<>();  
 Statement statement = connection.createStatement();  
 String sql = "select \* from Employees where ID = " + id;  
 ResultSet rs = statement.executeQuery(sql);  
 while (rs.next()) {  
 list.add(new Employees(rs.getLong("ID"), rs.getString("FirstName"), rs.getString("LastName"),  
 rs.getString("Phone"), rs.getBoolean("isAdmin"), rs.getString("Username"), rs.getString("Password")));  
 }  
 connection.close();  
 if (list.size() > 0) return list.get(0);  
 else return new Employees();  
 }  
  
 public long getAssetID(String SN) throws SQLException{  
 long id = 0;  
 Statement statement = connection.createStatement();  
 String sql = "select ID from Assets where AssetSN = '" + SN + "'";  
 ResultSet rs = statement.executeQuery(sql);  
 while (rs.next()){  
 id = rs.getLong("ID");  
 if (id > 0){  
 connection.close();  
 return id;  
 }  
 }  
 connection.close();  
 return id;  
 }  
  
 public boolean AssetIDisExists(long ID) throws SQLException{  
 Statement statement = connection.createStatement();  
 String sql = "select \* from Assets where ID = " + ID;  
 ResultSet rs = statement.executeQuery(sql);  
 while (rs.next()) {  
 long id = 0;  
 id = rs.getLong("ID");  
 if (id > 0) {  
 connection.close();  
 return true;  
 }  
 }  
 connection.close();  
 return false;  
 }  
  
 public Assets getAsset(long ID) throws SQLException{  
 Statement statement = connection.createStatement();  
 String sql = "select \* from Assets where ID = " + ID;  
 ResultSet rs = statement.executeQuery(sql);  
 if(rs.next()){  
 Assets assets = new Assets(rs.getLong("ID"), rs.getString("AssetSN"), rs.getString("AssetName"), rs.getLong("DepartmentLocationID"),  
 rs.getLong("EmployeeID"), rs.getLong("AssetGroupID"), rs.getString("Description")+"", rs.getString("WarrantyDate"));  
 connection.close();  
 return assets;  
 }  
 connection.close();  
 return new Assets();  
 }  
  
 public boolean deleteAsset(Assets assets) throws SQLException{  
 Statement statement = connection.createStatement();  
 String sql = "delete from Assets where ID = " + assets.getID();  
 String sqlPT = "delete from AssetPhotos where AssetID = " + assets.getID();  
 try {  
 statement.execute(sqlPT);  
 statement.execute(sql);  
 }catch (SQLException e){  
 return false;  
 }  
 return true;  
 }  
  
 public void insertPhoto(byte[] img, long AssetID) throws SQLException{  
 PreparedStatement preparedStatement = connection.prepareStatement("insert into AssetPhotos (AssetID, AssetPhoto) values (?,?)");  
 preparedStatement.setLong(1, AssetID);  
 preparedStatement.setBytes(2, img); *// <- lưu ảnh vào csdl dang byte* preparedStatement.execute();  
 connection.close();  
 }  
  
 public void updatePhoto(long ID, byte[] img) throws SQLException{  
 PreparedStatement preparedStatement = connection.prepareStatement("update AssetPhotos set AssetPhoto = ? where ID = ?");  
 preparedStatement.setBytes(1, img); *// <- lưu ảnh vào csdl dang byte* preparedStatement.setLong(2, ID);  
 preparedStatement.execute();  
 connection.close();  
 }  
  
 public byte[] getPhoto(long AssetID, int row) throws SQLException{  
 Statement statement = connection.createStatement();  
 ResultSet resultSet = statement.executeQuery("select \* from AssetPhotos where AssetID = "+AssetID);  
 for (int i = 0; i < row; i++) {  
 resultSet.next();  
 }  
 byte[] bytesImageDB = resultSet.getBytes(3); *// <- ảnh nằm ở cột 3* return bytesImageDB;  
 }  
  
 public long getPhotoID(byte[] img, long assetID) throws SQLException{  
 PreparedStatement preparedStatement = connection.prepareStatement("select ID from AssetPhotos where AssetPhoto = ? and AssetID = ?");  
 preparedStatement.setBytes(1, img);  
 preparedStatement.setLong(2, assetID);  
 ResultSet resultSet = preparedStatement.executeQuery();  
 if (resultSet.next()){  
 long id = resultSet.getLong("ID");  
 connection.close();  
 return id;  
 }  
 return 0;  
 }  
  
 public int getCountPhoto(long AssetID) throws SQLException{  
 Statement statement = connection.createStatement();  
 ResultSet resultSet = statement.executeQuery("select count(ID) as COUNT\_COLUMN from AssetPhotos where AssetID = "+AssetID);  
 resultSet.next();  
 int count = resultSet.getInt("COUNT\_COLUMN");  
 connection.close();  
 return count;  
 }  
  
 public void insertAssets(Assets assets) throws SQLException{  
 Statement statement = connection.createStatement();  
 PreparedStatement preparedStatement = connection.prepareStatement("insert into Assets (AssetSN, AssetName, DepartmentLocationID, EmployeeID, AssetGroupID, Description, WarrantyDate) values (?,?,?,?,?,?,?)");  
 preparedStatement.setString(1, assets.getAssetSN());  
 preparedStatement.setString(2, assets.getAssetName());  
 preparedStatement.setLong(3, assets.getDepartmentLocationID());  
 preparedStatement.setLong(4, assets.getEmployeeID());  
 preparedStatement.setLong(5, assets.getAssetGroupID());  
 preparedStatement.setString(6, assets.getDescription());  
 preparedStatement.setString(7, assets.getWarrantyDate());  
 preparedStatement.execute();  
 connection.close();  
 }  
  
 public void updateAssets(Assets assets) throws SQLException{  
 Statement statement = connection.createStatement();  
 PreparedStatement preparedStatement = connection.prepareStatement("update Assets set AssetSN = ?, AssetName = ?, DepartmentLocationID = ?, EmployeeID = ?," +  
 "AssetGroupID = ?, Description = ?, WarrantyDate = ? where ID = ?");  
 preparedStatement.setString(1, assets.getAssetSN());  
 preparedStatement.setString(2, assets.getAssetName());  
 preparedStatement.setLong(3, assets.getDepartmentLocationID());  
 preparedStatement.setLong(4, assets.getEmployeeID());  
 preparedStatement.setLong(5, assets.getAssetGroupID());  
 preparedStatement.setString(6, assets.getDescription());  
 preparedStatement.setString(7, assets.getWarrantyDate());  
 preparedStatement.setLong(8, assets.getID());  
 preparedStatement.execute();  
 connection.close();  
 }  
  
 public void moveAssetDepartmentTo(long dep\_loc\_ID, String AssetSN, long assID) throws SQLException{  
 Statement statement = connection.createStatement();  
 PreparedStatement preparedStatement = connection.prepareStatement("update Assets set AssetSN = ?, DepartmentLocationID = ? where ID = ?");  
 preparedStatement.setString(1, AssetSN);  
 preparedStatement.setLong(2, dep\_loc\_ID);  
 preparedStatement.setLong(3, assID);  
 preparedStatement.execute();  
 connection.close();  
 }  
}