Tìm kiếm sản phẩm

**public** List<Product> searchByName(String txtSearch) {

List<Product> list = **new** ArrayList<>();

String query = " SELECT \* from dbo.fn\_SearchProductName(?)";

**try** {

conn = **new** DBContext().getConnection();

ps = conn.prepareStatement(query);

ps.setString(1, txtSearch);

rs = ps.executeQuery();

**while** (rs.next()) {

list.add(**new** Product(rs.getInt(1),

rs.getString(2),

rs.getString(3),

rs.getDouble(4),

rs.getString(5),

rs.getString(6)));

}

} **catch** (Exception e) {

}

**return** list;

}

**Load sản phẩm và phân trang**

**public** List<Product> pagingProduct(**int** index) {

List<Product> list = **new** ArrayList<>();

**try** {

conn = **new** DBContext().getConnection();//mo ket noi voi sql

CallableStatement command = conn.prepareCall("{call sp\_PagingProduct (?) }");

command.setInt(1, (index-1)\*6);

ResultSet rs = command.executeQuery();

ps = conn.prepareStatement(query);

ps.setInt(1,(index-1)\*6);

rs = ps.executeQuery();

**while** (rs.next()) {

list.add(**new** Product(rs.getInt(1),

rs.getString(2),

rs.getString(3),

rs.getDouble(4),

rs.getString(5),

rs.getString(6)));

}

} **catch** (Exception e) {

}

**return** list;

}

**//Load và phân trang account trong manage account**

**public** List<Account> pagingAccount(Account account,**int** index) {

List<Account> list = **new** ArrayList<>();

String query = "select \* from Account\n"

+"ORDER BY uid\n"

+"OFFSET ? ROWS FETCH NEXT 6 ROWS ONLY;";

**try** {

conn = **new** DBContext().getConnection(account.getUser(),account.getPass());

ps = conn.prepareStatement(query);

ps.setInt(1,(index-1)\*6);

rs = ps.executeQuery();

**while** (rs.next()) {

list.add(**new** Account(rs.getInt(1),

rs.getString(2),

rs.getString(3),

rs.getInt(4),

rs.getInt(5)

));

}

} **catch** (Exception e) {

}

**return** list;

}

**Thêm sản phẩm**

**public** **void** insertProduct(Account account,String name, String image, String price,String title, String description, String category, **int** sid) {

**try** {

conn = **new** DBContext().getConnection(account.getUser(),account.getPass());

CallableStatement stmt = conn.prepareCall("{call sp\_InsertProduct(?,?,?,?,?,?,?) }");

stmt.setString(1, name);

stmt.setString(2, image);

stmt.setString(3, price);

stmt.setString(4, title);

stmt.setString(5, description);

stmt.setString(6, category);

stmt.setInt(7, sid);

stmt.execute();

} **catch** (Exception e) {

}

}

**Sửa sản phẩm**

**public** **void** editProduct(Account account,String name, String image, String price,

String title, String description, String category, String pid, **int** sid) {

**try** {

conn = **new** DBContext().getConnection(account.getUser(),account.getPass());

CallableStatement stmt = conn.prepareCall("{call sp\_UpdateProduct(?,?,?,?,?,?,?,?) }");

stmt.setString(1, pid);

stmt.setString(2, name);

stmt.setString(3, image);

stmt.setString(4, price);

stmt.setString(5, title);

stmt.setString(6, description);

stmt.setString(7, category);

stmt.setInt(8, sid);

stmt.execute();

} **catch** (Exception e) {

}

}

**Xóa sản phẩm**

**public** **void** deleteProduct(Account account,String pid) {

**try** {

conn = **new** DBContext().getConnection(account.getUser(),account.getPass());

CallableStatement stmt = conn.prepareCall("{call sp\_DeleteProduct(?) }");

stmt.setString(1, pid);

stmt.execute();

}

**catch** (Exception e) {

}

}

**Load sản phẩm theo category**

**public** List<Product> getProductByCID(String cid) {

List<Product> list = **new** ArrayList<>();

String query = "select \* from Product\n" +"Where cateID =?";

**try** {

conn = **new** DBContext().getConnection();

ps = conn.prepareStatement(query);

ps.setString(1, cid);

rs = ps.executeQuery();

**while** (rs.next()) {

list.add(**new** Product(rs.getInt(1),

rs.getString(2),

rs.getString(3),

rs.getDouble(4),

rs.getString(5),

rs.getString(6)));

}

} **catch** (Exception e) {

}

**return** list;

}

**Xem chi tiết sản phẩm**

**public** Product getProductByID(String pid) {

String query = "select \* from Product\n" +"Where id =?";

**try** {

conn = **new** DBContext().getConnection();//mo ket noi voi sql

ps = conn.prepareStatement(query);

ps.setString(1, pid);

rs = ps.executeQuery();

**while** (rs.next()) {

**return** **new** Product(rs.getInt(1),

rs.getString(2),

rs.getString(3),

rs.getDouble(4),

rs.getString(5),

rs.getString(6));

}

} **catch** (Exception e) {

}

**return** **null**;

}

**Load all category**

**public** List<Category> getAllCategory() {

List<Category> list = **new** ArrayList<>();

**try** {

conn = **new** DBContext().getConnection();//mo ket noi voi sql

CallableStatement command = conn.prepareCall("{call sp\_getAllCategory }");

ResultSet rs = command.executeQuery();

**while** (rs.next()) {

list.add(**new** Category(rs.getInt(1),

rs.getString(2)));

}

} **catch** (Exception e) {

}

**return** list;

}

**Load last product**

**public** Product getLast() {

**try** {

conn = **new** DBContext().getConnection();//mo ket noi voi sql

CallableStatement command = conn.prepareCall("{call sp\_getLastProduct }");

ResultSet rs = command.executeQuery();

**while**(rs.next()){

**return** **new** Product(rs.getInt(1),

rs.getString(2),

rs.getString(3),

rs.getDouble(4),

rs.getString(5),

rs.getString(6));

}

} **catch** (Exception e) {

}

**return** **null**;

}

**thêm account**

**public** **void** insertAccount(Account account,String user, String pass, String isSell,String isAdmin) {

**try** {

conn = **new** DBContext().getConnection();

CallableStatement stmt = conn.prepareCall("{call sp\_InsertAccount(?,?,?,?) }");

stmt.setString(1, user);

stmt.setString(2, pass);

stmt.setString(3, isSell);

stmt.setString(4, isAdmin);

stmt.execute();

} **catch** (Exception e) {

e.printStackTrace();

}

}

**//sửa account**

**public** **void** editAccount(Account account, String user, String pass,

String isSell, String isAdmin,String uid) {

**try** {

conn = **new** DBContext().getConnection(account.getUser(),account.getPass());//mo ket noi voi sql

CallableStatement stmt = conn.prepareCall("{call sp\_UpdateAccount(?,?,?,?,?) }");

stmt.setString(1, uid);

stmt.setString(2, user);

stmt.setString(3, pass);

stmt.setString(4, isSell);

stmt.setString(5, isAdmin);

stmt.execute();

} **catch** (Exception e) {

}

}

**Xóa account**

**public** **void** deleteAccount(Account account,String uid) {

**try** {

conn = **new** DBContext().getConnection(account.getUser(),account.getPass());

CallableStatement stmt = conn.prepareCall("{call sp\_DeleteAccount(?) }");

stmt.setString(1, uid);

stmt.execute();

}

**catch** (Exception e) {

}

}

**//đăng nhập**

**public** Account login(String user,String pass) {

String query = "select \* from Account\n" +"Where [user] = ?\n"+"and pass = ?";

**try** {

conn = **new** DBContext().getConnection();//mo ket noi voi sql

ps = conn.prepareStatement(query);

ps.setString(1,user);

ps.setString(2,pass);

rs = ps.executeQuery();

**while** (rs.next()) {

**return** **new** Account(rs.getInt(1),

rs.getString(2),

rs.getString(3),

rs.getInt(4),

rs.getInt(5));

}

}

**catch** (Exception e) {

}

**return** **null**;

}

**//kiểm tra tài khoản**

**public** Account checkAccountExist(String user) {

String query = "select \* from Account\n" +"Where [user] = ?";

**try** {

conn = **new** DBContext().getConnection();//mo ket noi voi sql

ps = conn.prepareStatement(query);

ps.setString(1,user);

rs = ps.executeQuery();

**while** (rs.next()) {

**return** **new** Account(rs.getInt(1),

rs.getString(2),

rs.getString(3),

rs.getInt(4),

rs.getInt(5));

}

}

**catch** (Exception e) {

}

**return** **null**;

}

**//đổi mật khẩu**

**public** **void** editPassWord(String pass , **int** uid) {

String query = "update Account\n"

+ "set [pass] = ?\n"

+"where uid = ?";

**try** {

conn = **new** DBContext().getConnection();//mo ket noi voi sql

ps = conn.prepareStatement(query);

ps.setString(1, pass);

ps.setInt(2, uid);

ps.executeUpdate();

} **catch** (Exception e) {

}

}

**//Thêm thông tin sản phẩm vào bảng Order và Detail Order**

**public** **void** saveOrder(String uid, List<Product> list) {

**try** {

conn = **new** DBContext().getConnection();//mo ket noi voi sql

//1.add Order to database

CallableStatement stmt = conn.prepareCall("{call sp\_InsertOrderWithDetail (?) }");

stmt.setInt(1,Integer.*parseInt*(uid));

stmt.execute();

//2.lấy oid Order vừa tạo

**int** lastID = *getLastOrderID*();

//3.add DetailOrder

**for**(Product p: list){

insertDetailOrder(String.*valueOf*(lastID),String.*valueOf*(p.getId()), String.*valueOf*(p.getAmount()));

}

} **catch** (Exception e) {

}

}

**//lấy oid Order vừa tạo**

**public** **static** **int** getLastOrderID() {

**try** {

Connection conn = **new** DBContext().getConnection();//mo ket noi voi sql

//2.lấy oid Order vừa tạo

CallableStatement stmt = conn.prepareCall("{?= call fn\_LastOrderID()}");

stmt.registerOutParameter(1,Types.***INTEGER***);

stmt.execute();

**return** stmt.getInt(1);

} **catch** (Exception e) {

e.getStackTrace();

}

**return** -1;

}

**//Lấy ra order vừa tạo**

**public** Order getLastOrder(){

Order order = **new** Order();

**try** {

Connection conn = **new** DBContext().getConnection();//mo ket noi voi sql

CallableStatement stmt = conn.prepareCall("{call sp\_getOneOrder(?) }");

**int** lastID = *getLastOrderID*();

stmt.setInt(1,lastID);

rs = stmt.executeQuery();

**while** (rs.next()) {

**return** **new** Order(rs.getInt(1),

rs.getInt(2),

rs.getInt(3),

rs.getDouble(4),

rs.getDate(5),

rs.getString(6),

rs.getString(7),

rs.getString(8),

rs.getString(9));

}

}**catch** (Exception e) {

e.getStackTrace();

}

**return** **null**;

}

**//xem thông tin trong view ThongTinSanPham**

**public** List<Product> getViewProduct() {

List<Product> list = **new** ArrayList<>();

String query = "select \* from ThongTinSanPham";

**try** {

conn = **new** DBContext().getConnection();//mo ket noi voi sql

ps = conn.prepareStatement(query);

rs = ps.executeQuery();

**while** (rs.next()) {

list.add(**new** Product(rs.getString(1),

rs.getString(2),

rs.getDouble(3),

rs.getString(4)

));

}

} **catch** (Exception e) {

}

**return** list;

}

**//Lấy những product mà do người đó bán để load lên trong manage product**

**public** List<Product> getProductBySaleID(Account account,**int** id) {

List<Product> list = **new** ArrayList<>();

**try** {

conn = **new** DBContext().getConnection(account.getUser(),account.getPass());

CallableStatement stmt = conn.prepareCall("{call sp\_getProductBySaleID(?) }");

stmt.setInt(1,id);

rs = stmt.executeQuery();

**while** (rs.next()) {

list.add(**new** Product(rs.getInt(1),

rs.getString(2),

rs.getString(3),

rs.getDouble(4),

rs.getString(5),

rs.getString(6)));

}

} **catch** (Exception e) {

}

**return** list;

}

**//Đăng ký tài khoản**

**public** **void** signup(String user, String pass) {

**try** {

conn = **new** DBContext().getConnection();//mo ket noi voi sql

CallableStatement stmt = conn.prepareCall("{call sp\_InsertAccount(?,?,0,0) }");

stmt.setString(1,user);

stmt.setString(2,pass);

stmt.execute();

}

**catch** (Exception e) {

}

}