



PROJECT REPORT

Mini Supermarkets

Semester : Programming Fundamentals

Class : PF08

Group : Group 4

Instructor : Nguyễn Xuân Sinh

Team Members : NDE19021_Nguyễn Quyết Thắng

NDE19064_Nguyễn Vũ Huân

NDE19020_Trần Văn Huân

Index

Project Name	1
Index	2
I. Project introduction	3
II. Analyze System Requirements	4
III. Design Details	17
IV. Test	45
V. Assign work to each team member	56
VI. Installation Instructions	59
VII. Appendix	66

I. Project Introduction

Process payment for orders in mini supermarkets

1. Proposed System

Actor Manager

Actor Casher

Manager function

1. Product management
2. Accounts Management
3. Orders Management

Cashier function

1. Create order

2. The scope of the project to be applied

Checkout and update invoice

Create and update product

3. System Name

Mini supermarkets management system

4. Deployment Environment

JDK13

5. Development Tools

Visual Studio Code

MySQL WorkBench

IntelliJ IDEA

Window 10

JDK13

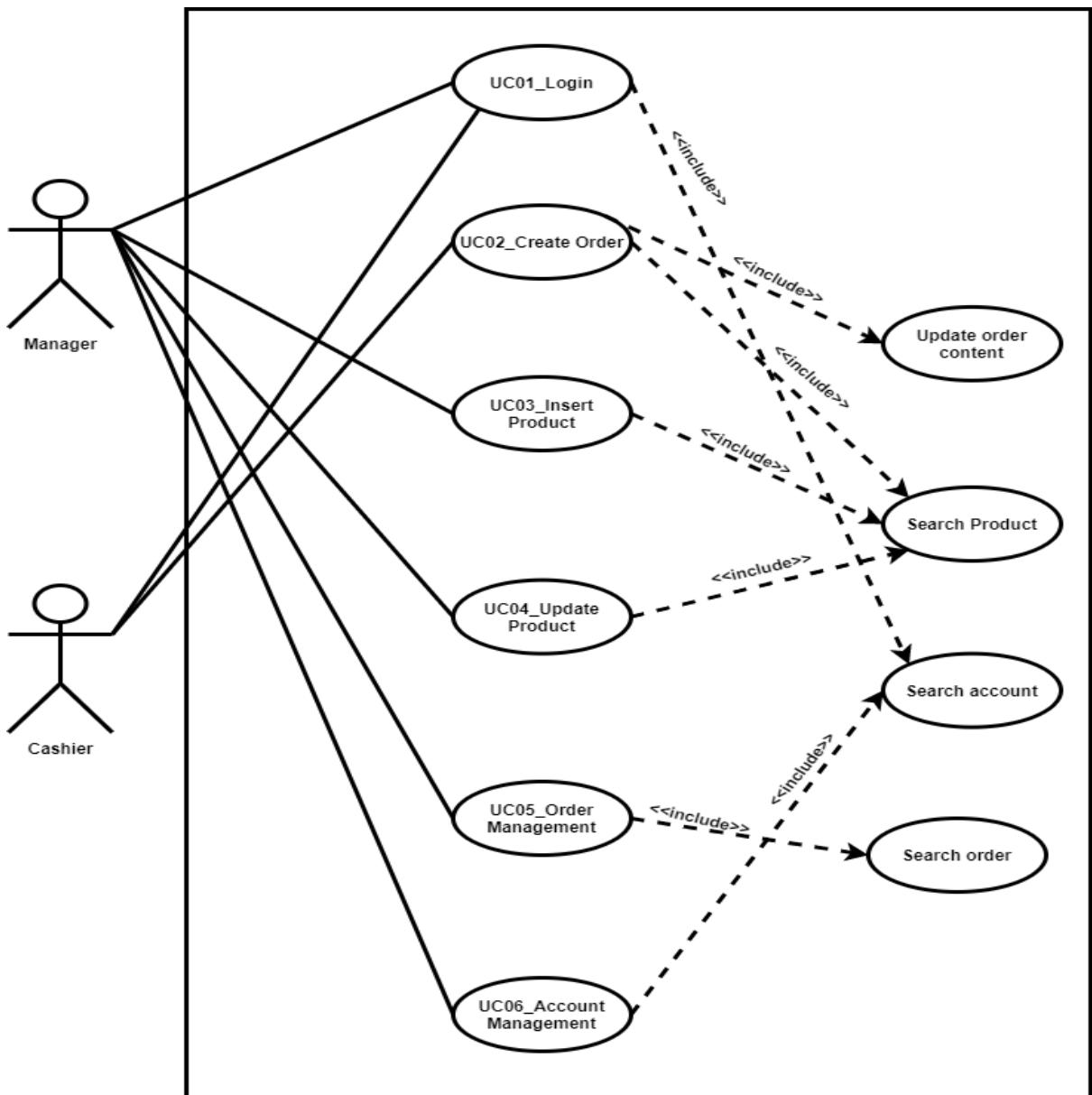
6. Customer Requirements

- The customer needs a system with the following functions:

- + Management products (quantity, price, information) of products in store
- + Export sales invoices and statistics on the amount of invoices issued
- + Security by event login
- + Divide separate functions for managers and employees
- + Create and management account of cashiers

II. Analyze System Requirements

1. Use case

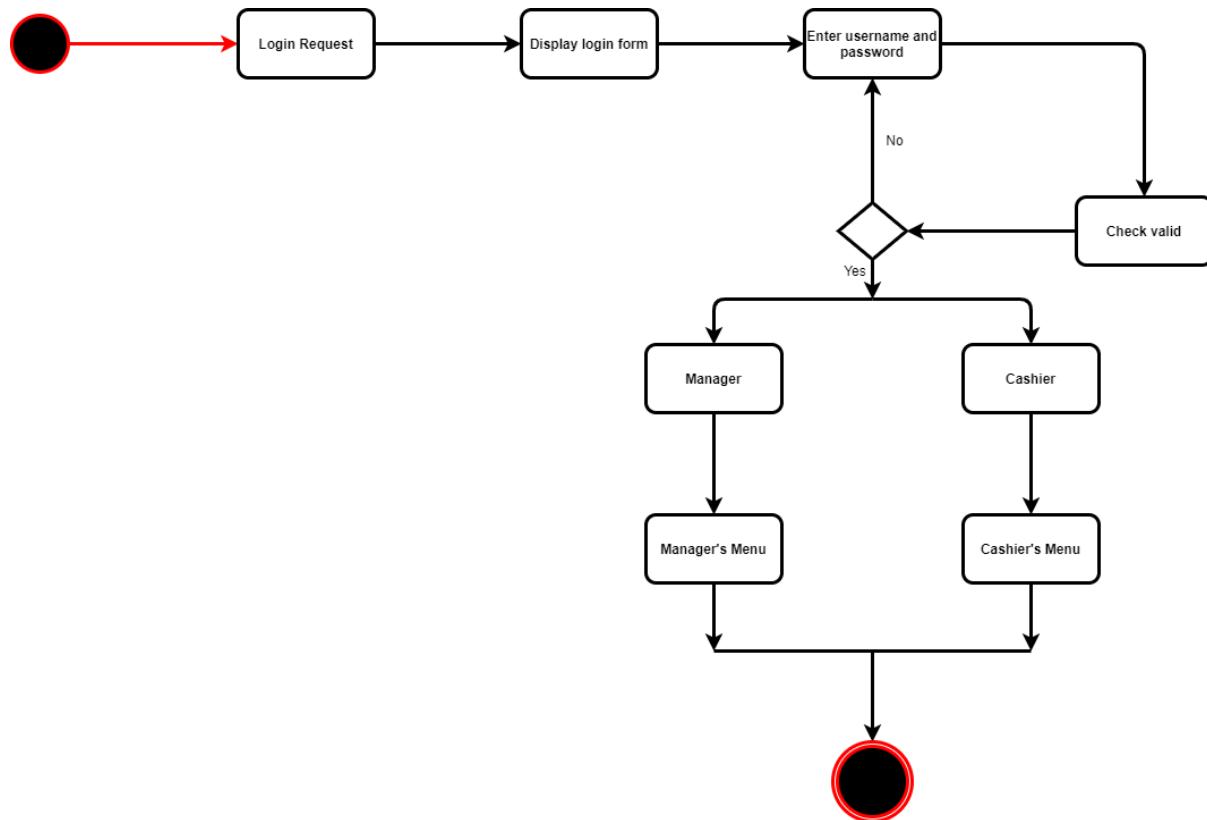


Use Case diagram

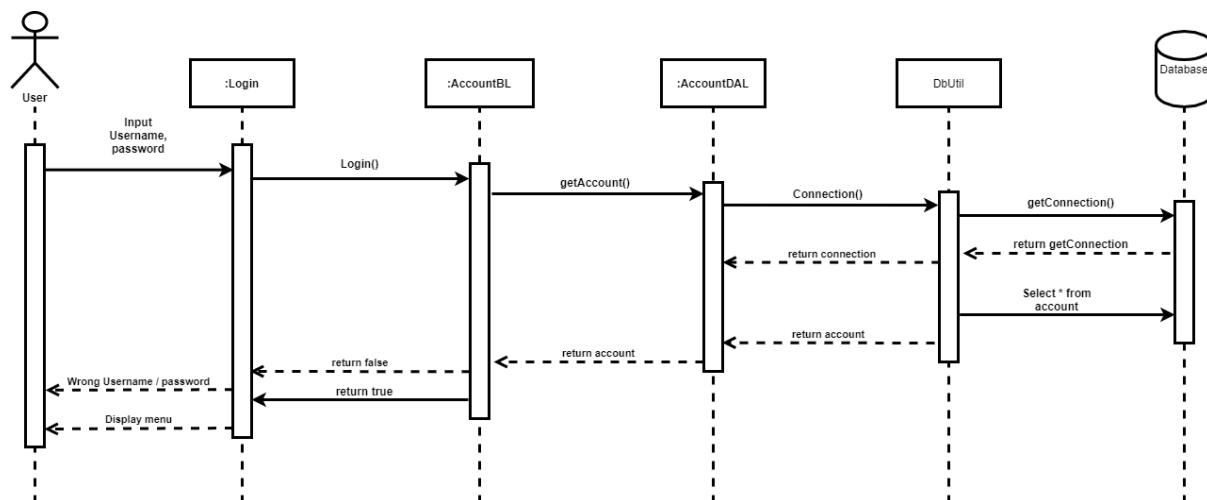
UC01_Login (Description)

Name	Login
ID	UC01
Description	Manager and Cashier must login to system
Actor	Manager and Cashier
Organizational Benefits	Authentication
Frequency of Use	Often
Triggers	When a user wants to use the system
Preconditions	
Postconditions	Display functions of the cashier /manager
Main Course	1. User input user name and password 2. Valid user name/password 3. Display Functions of the user 3.1. If user is Manager (Manager's Menu) (AC3.1) 3.2. If user is Cashier (Cashier's Menu) (AC3.2)
Alternate Courses	AC3.1: Manager's Menu 1. Display: - Products Management - Orders Management - Accounts Management AC3.2: Cashier's Menu 2. Display: - Create order - Update order
Exceptions	EX1. Valid Username: - None special characters - Maximum 20 characters EX2. 2.1 Valid Password - Minimum password length 8 characters - Minimum 1 uppercase character - Minimum 1 normal character - Minimum 1 numeric character 2.2 Username Password fails - Return user to Main Course Step 1

UC01_Login (ActivityDiagram)



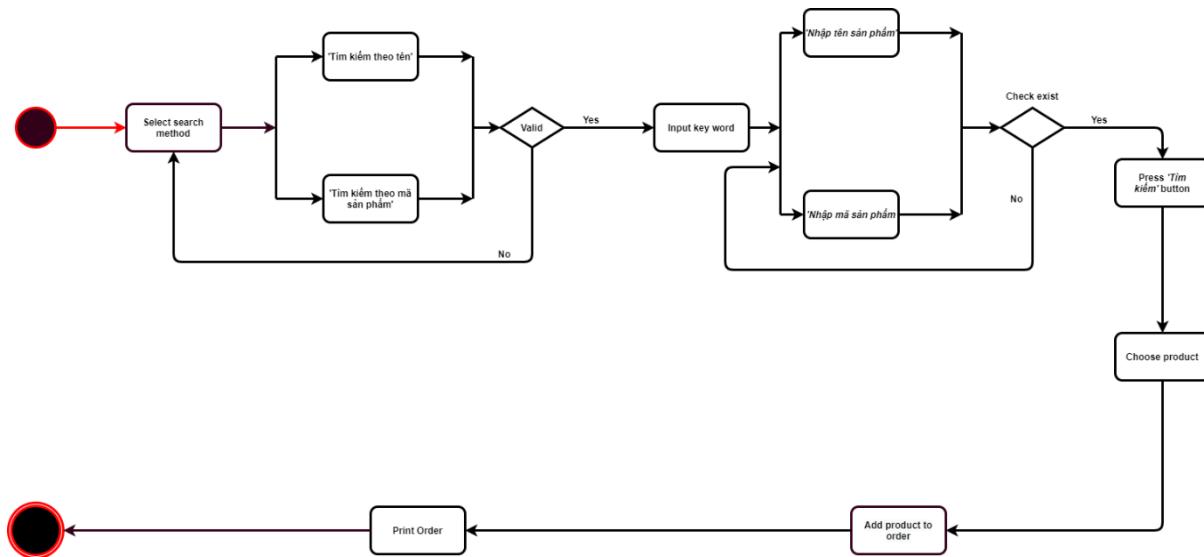
UC01_Login (Sequence Diagram)



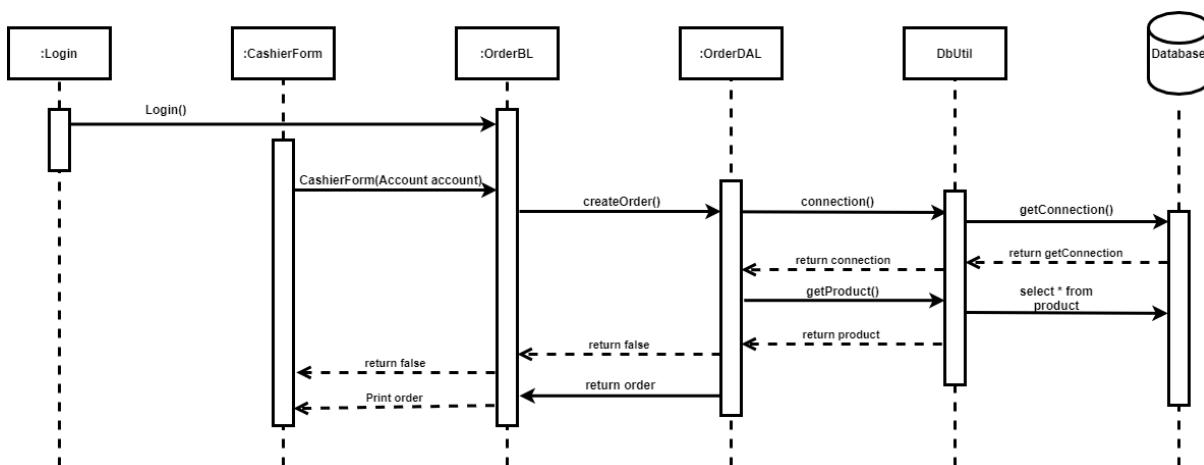
UC02_Create Order (Description)

Name	Create order
ID	UC02
Description	Create new order
Actor	Cashier
Organizational Benefit	Control action sale of the business
Frequency of Use	Often
Triggers	User selects the “Create Order” Function
Preconditions	User is Cashier was Login
Postconditions	Print Order
Main Courses	<ol style="list-style-type: none"> 1. Select search method (AC1) 2. Input key word (AC2) 3. Press ‘Tìm kiếm’ button 4. Choose product (AC4) 5. Add product to order (AC5) 6. Print order
Alternate Courses	<p>AC1.</p> <ol style="list-style-type: none"> 1. Search method <ol style="list-style-type: none"> 1.1.1 ‘Tìm kiếm theo tên’ 1.1.2 ‘Tìm kiếm theo mã sản phẩm’ 2. Valid search method <ol style="list-style-type: none"> 2.1.1 If user not search form yet <ul style="list-style-type: none"> - Prompt user: ‘Hãy chọn phương thức tìm kiếm’ <p>AC2:</p> <ol style="list-style-type: none"> 2.1 ‘Nhập tên sản phẩm’ 2.2 ‘Nhập mã sản phẩm’ <p>AC4:</p> <ol style="list-style-type: none"> 4.1. Click left mouse to product 4.2. Click right mouse to product <p>AC5:</p> <ol style="list-style-type: none"> 5.1: Press ‘Thêm vào hóa đơn’ button 5.2: Press ‘Yes’ button 5.3: Input ‘Số lượng’ 5.4: Press ‘ok’ button
Exception	<p>EX 1: Product doesn't exist</p> <ol style="list-style-type: none"> 1. System prompt user: Product doesn't exist in system <p>EX3: Quantity < = 0</p> <ol style="list-style-type: none"> 1. Return user Main Course Step 4

UC02_Create Order (Activity Diagram)



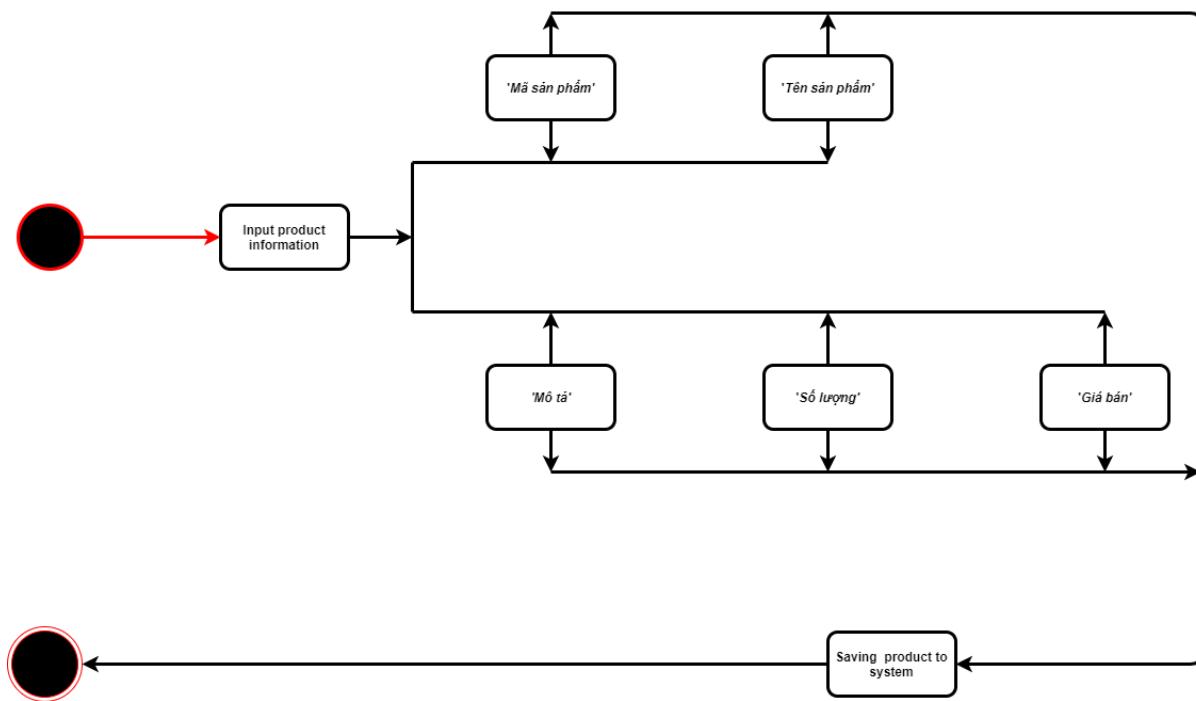
UC02_Create Order (SequenceDiagram)



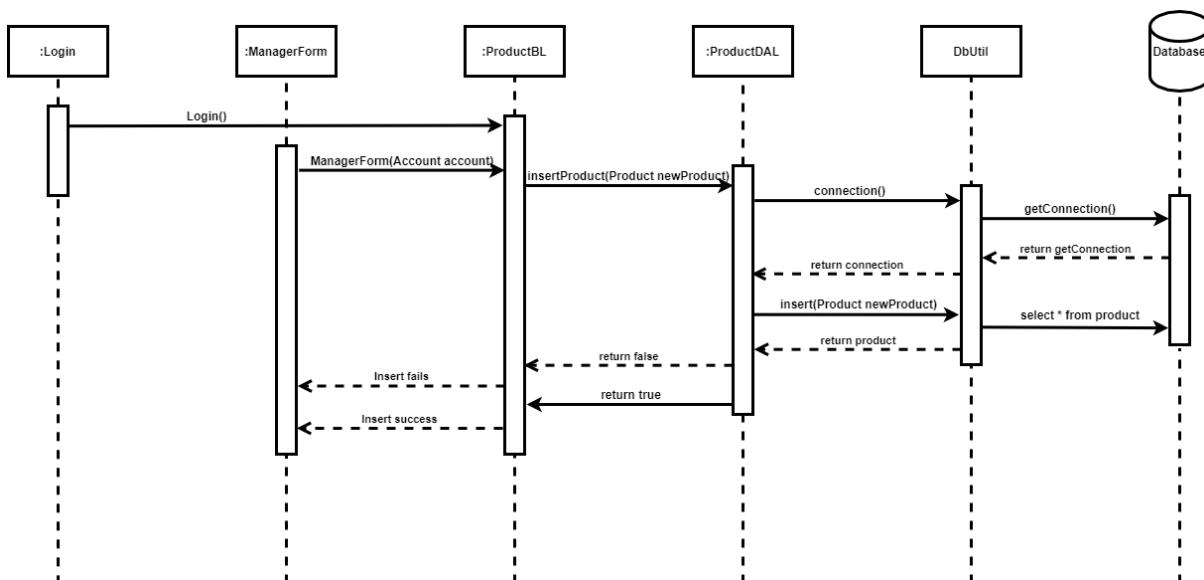
UC03_Update Order (Description)

Name	Insert Product
ID	UC03
Description	Add new product to system
Actor	Manager
Organizational Benefit	Manage products in store
Frequency of Use	Often
Triggers	Selects 'Thêm mới sản phẩm'
Preconditions	User is Manager
Postconditions	Prompt user 'Thêm sản phẩm thành công'
Main Courses	<ol style="list-style-type: none"> 1. Input product information (AC1) 2. Add product to system 3. Click 'ok' 4. Saving to system
Alternate Courses	<p>AC1: Input product information:</p> <ol style="list-style-type: none"> 1. Input 'Mã sản phẩm' <ul style="list-style-type: none"> - Click 'Kiểm tra' to check exist 2. Input 'Tên sản phẩm' <ul style="list-style-type: none"> - Click 'Kiểm tra' to check exist 3. Input 'Mô tả' 4. Input 'Số lượng' 5. Input 'Đơn vị' 6. Input 'Giá bán'
Exception	<p>EX1: Product does existed</p> <ol style="list-style-type: none"> 1. Return user to Main courses step 1 <p>EX2.1: Valid Name product</p> <ol style="list-style-type: none"> 1. Name does existed <ul style="list-style-type: none"> - Prompt user: 'Sản phẩm này đã tồn tại trong hệ thống' - Return Main course 2 <p>EX2.2: Valid 'Số lượng'</p> <ol style="list-style-type: none"> 3.1. Quantity > 0 <p>EX2.3: Valid 'Giá bán'</p> <ol style="list-style-type: none"> 2.1. Price > 0

UC03_Update Order (Activity Diagram)



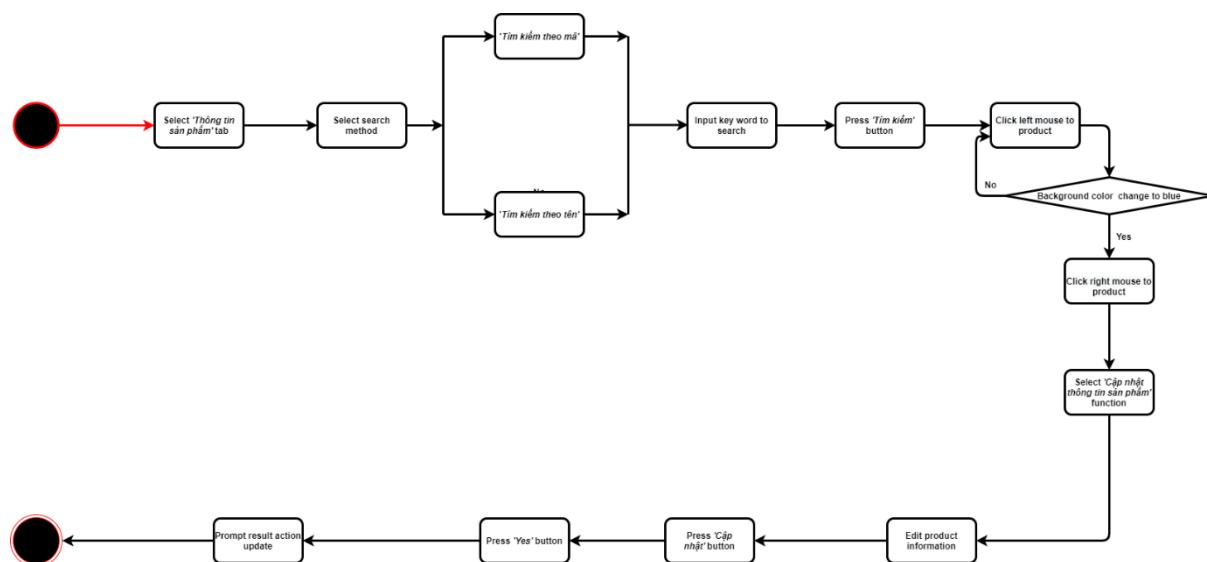
UC03_Update Order (SequenceDiagram)



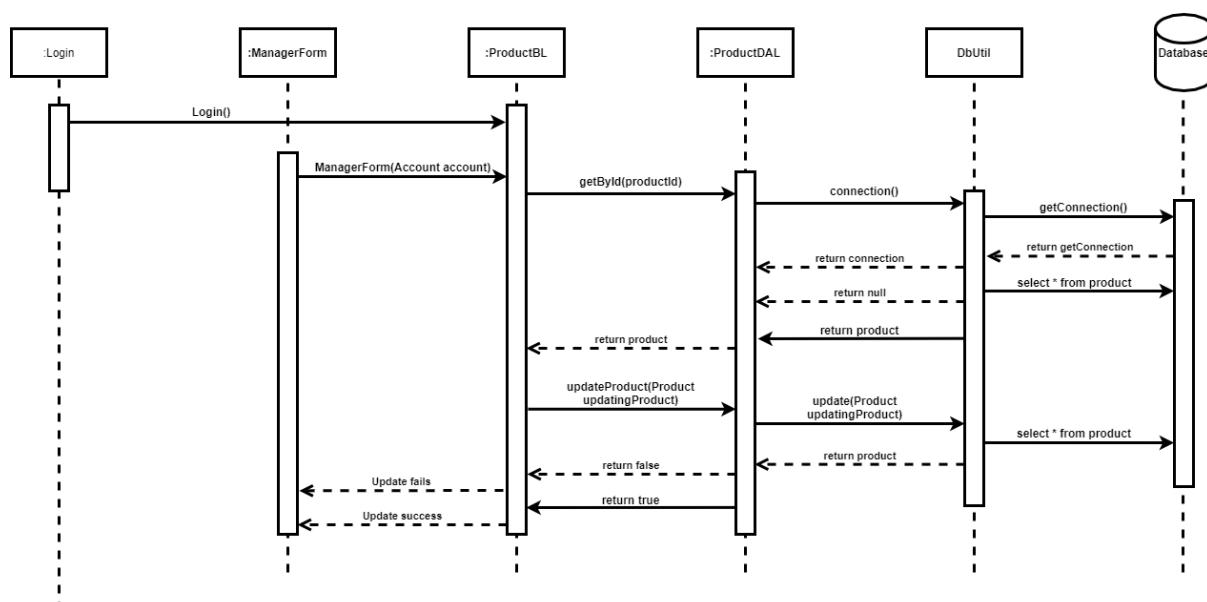
UC04_Insert Product (Description)

Name	Update Product
ID	UC04
Description	Update product information
Actor	Manager
Organizational Benefits	Update accurate product information
Frequency of Use	Sometime
Triggers	Select 'Quản lý sản phẩm' tab
Preconditions	User is Manager
Postcondition	System confirm update success
Main Course	<ol style="list-style-type: none"> 1. Select 'Thông tin sản phẩm' tab 2. Select search method 3. Input key word to search 4. Press 'Tim kiếm' button 5. Click left mouse to product (AC05) 6. Click right mouse to product 7. Select 'Cập nhật thông tin sản phẩm' function (AC07)(AC07.2) 8. Edit product information (AC08) 9. Press 'Cập nhật' button (AC09) 10. Press 'Yes' button (AC10) 11. Prompt result action update (AC11)
Alternate Courses	<p>AC05:</p> <ul style="list-style-type: none"> - Background color change to blue color (EX5) <p>AC7:</p> <ul style="list-style-type: none"> - Show confirm dialog - Press 'yes' button (EX07.1) - Display update information <p>AC07.2:</p> <ul style="list-style-type: none"> - Auto fill old information of that to corresponding fields <p>AC08:</p> <ul style="list-style-type: none"> - User can edit data on fields to update information <p>AC09:</p> <ul style="list-style-type: none"> - If user press 'Hủy' button <ul style="list-style-type: none"> + Cancel update action <p>AC10:</p> <ul style="list-style-type: none"> - If user press 'Hủy' or 'Cancel' button <ul style="list-style-type: none"> + Cancel update action <p>AC11:</p> <ul style="list-style-type: none"> - If update success <ul style="list-style-type: none"> + Prompt: 'Cập nhật thành công' - If update fails <ul style="list-style-type: none"> + Prompt: 'Cập nhật thất bại'
Exceptions	<p>EX05:</p> <ul style="list-style-type: none"> If background color wasn't change to blue <ul style="list-style-type: none"> - Prompt: 'Vui lòng chọn sản phẩm' <p>EX07.1:</p> <ul style="list-style-type: none"> If user press 'no' button <ul style="list-style-type: none"> + Cancel update action

UC04_ Insert Product (Activity Diagram)



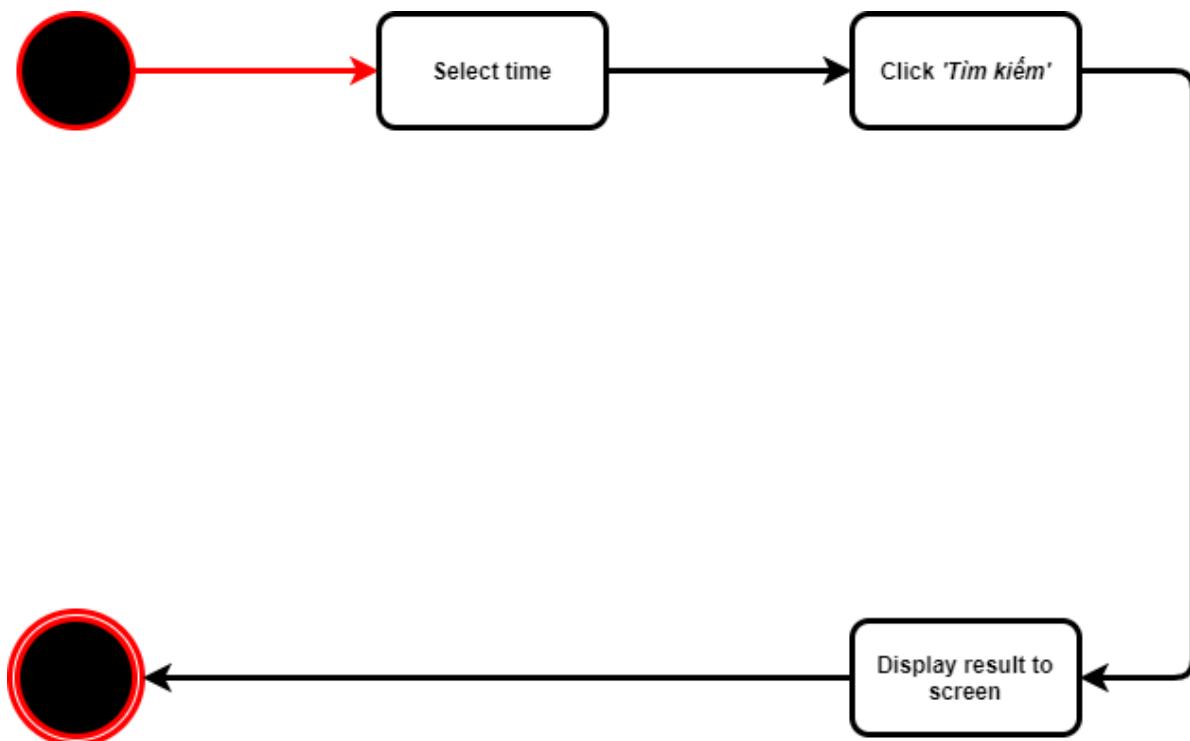
UC04_Insert Product (SequenceDiagram)



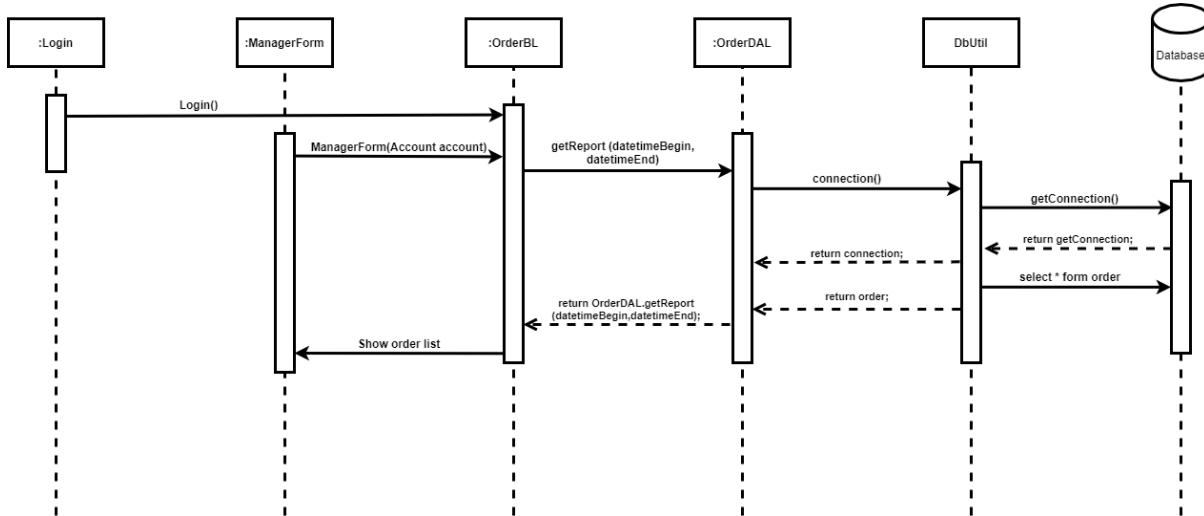
UC05_Update Product (Description)

Name	Management Order
ID	UC05
Description	Statistics order list
Actor	Manager
Organizational Benefit	Manage order in store
Frequency of Use	Often
Triggers	Selects 'Quản lý hóa đơn'
Preconditions	User is Manager
Postconditions	Show order list
Main Courses	<ol style="list-style-type: none"> 1. Select time (AC1) 2. Click 'Tìm kiếm' 3. Display results to screen (AC3)
Alternate Courses	<p>AC1: Select time</p> <ol style="list-style-type: none"> 1. In field begin day + Set day, month and year 2. In field End day 2.1: Set day, month and year 2.2: User press check box to get real time <p>AC3:</p> <ol style="list-style-type: none"> 1. User select order to see details (See UC04_AC05)
Exception	<p>EX2.2:</p> <ul style="list-style-type: none"> - If user press check box to get real time + Not allow to user choose end date time themself

UC05_Update Product (Activity Diagram)



UC05_Update Order(SequenceDiagram)

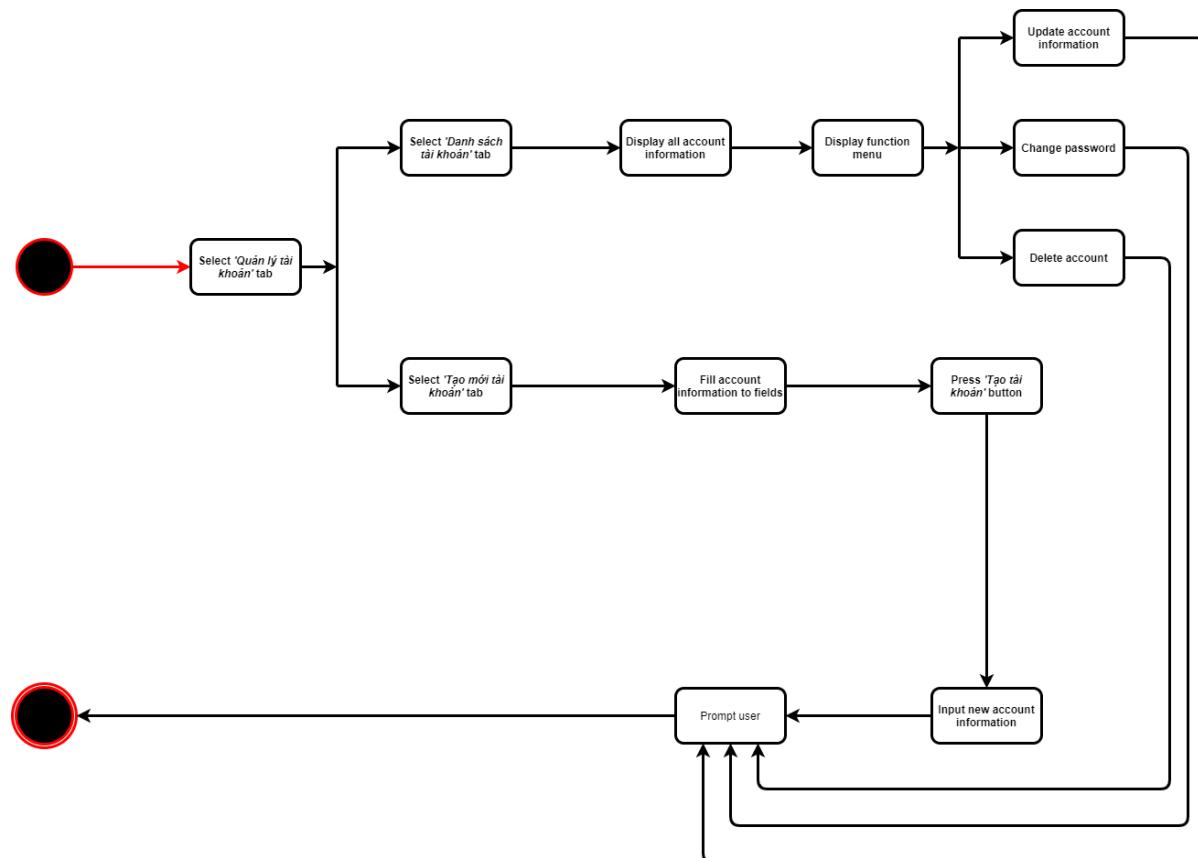


UC06_ Management Account (Description)

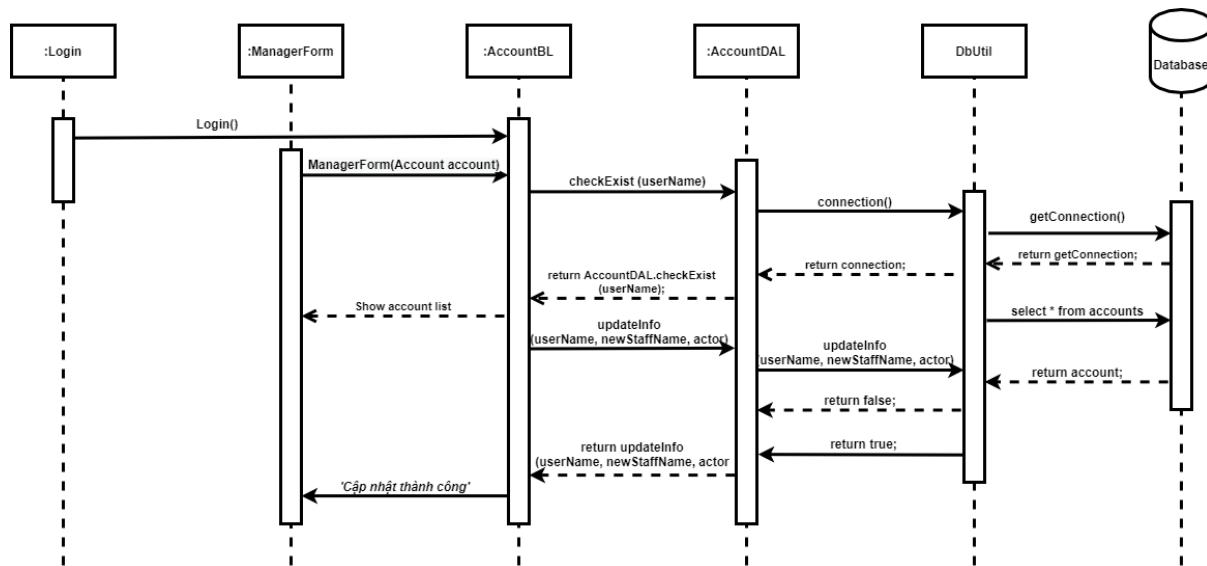
Name	Management account
ID	UC06
Description	Statistics account list
Actor	Manager
Organizational Benefit	Manage order in store
Frequency of Use	Often
Triggers	Selects 'Quản lý tài khoản'
Preconditions	User is Manager
Postconditions	Show account list
Main Courses	<ul style="list-style-type: none"> + 'Danh sách tài khoản' tab (AC1) (AC1.1)(AC1.1.2) + 'Thêm mới tài khoản' tab (AC2)
Alternate Courses	<p>AC1:</p> <ol style="list-style-type: none"> 1. Display all accounts information to screen <p>AC1.1: Display function menu</p> <ol style="list-style-type: none"> 1. See UC04_AC05 <p>AC1.1.1: Update accounts information</p> <ol style="list-style-type: none"> 1. Display update dialog 2. Fill information to corresponding fields 3. Press 'Cập nhật' button <p>AC1.1.2: Change Password</p> <ol style="list-style-type: none"> 1. Display confirm dialog (EX1.1.2.1) <p>AC1.1.3: Delete account</p> <ol style="list-style-type: none"> 1. Display confirm dialog <p>AC2: Select 'Thêm mới tài khoản' tab</p> <ol style="list-style-type: none"> 1. Fill account information to fields 2. Press 'Tạo tài khoản' button (EX2.2)
Exception	<p>EX1.1.1.3:</p> <ul style="list-style-type: none"> - User press 'Hủy' button <ul style="list-style-type: none"> + Cancel update action <p>EX1.1.2.1:</p> <ul style="list-style-type: none"> - User press 'No' or 'Cancel' button <ul style="list-style-type: none"> + Cancel action - User press 'Yes' button <ul style="list-style-type: none"> + Display input dialog If account changing is staff account <ul style="list-style-type: none"> ➔ User input new password

	<p>➔ Confirm</p> <p>If account changing is manager account</p> <p>➔ User input old password + Valid it</p> <p>➔ User input new password</p> <p>➔ Confirm</p> <p>EX1.1.3.1:</p> <ul style="list-style-type: none"> - User press 'No' or 'Cancel' button + Cancel action - User press 'Yes' button + Delete account + Prompt result action <p>EX2.2:</p> <ul style="list-style-type: none"> - User press 'Hủy' button + Clear all fields + Cancel action
--	---

UC06_ Management Account (Activity Diagram)



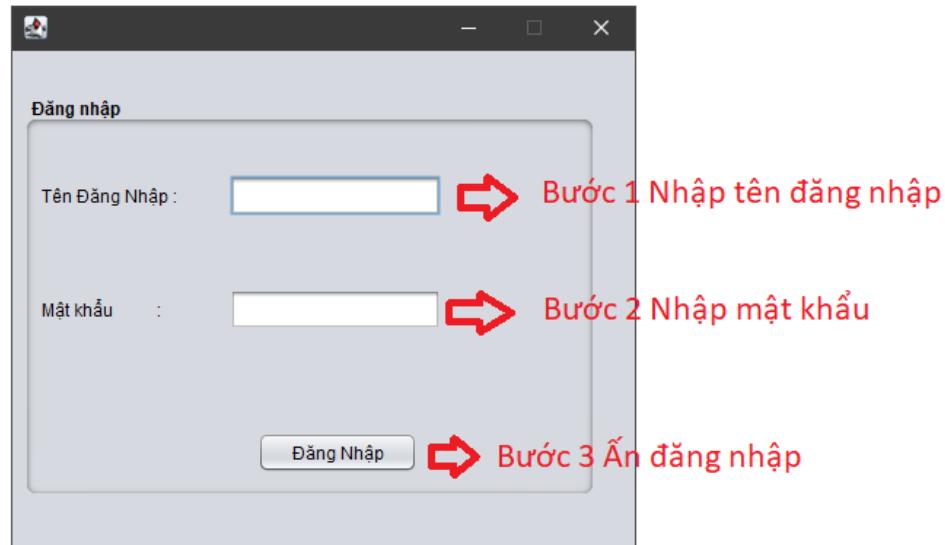
UC06_ Management Account (SequenceDiagram)

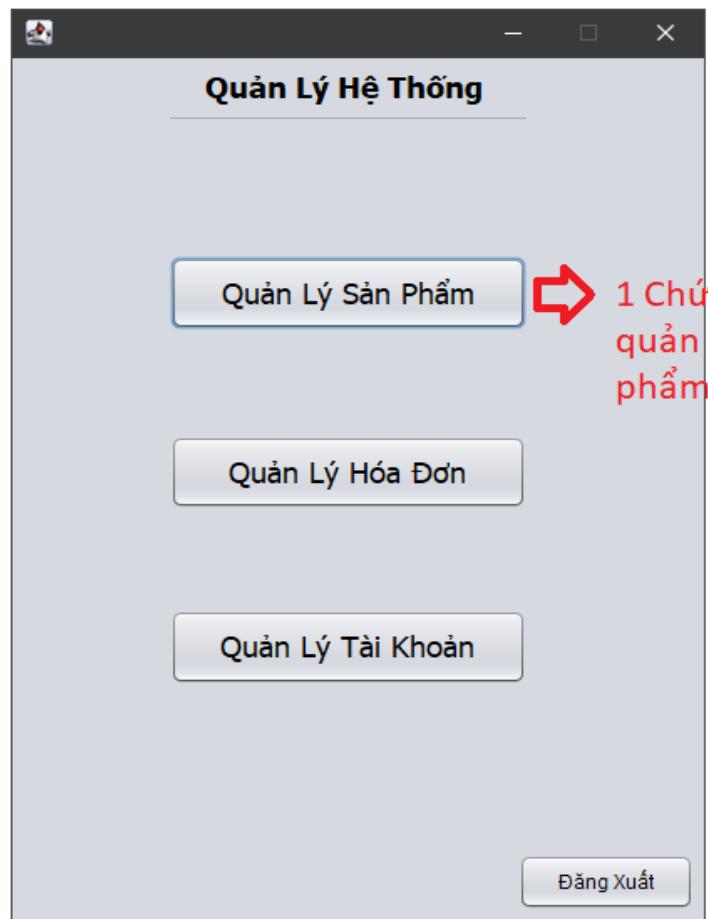


III. Design Details

1. Manager

Chức Năng đăng nhập





Giao diện quản lý sản phẩm

< Trở Lại

Quản Lý Sản Phẩm

1.1 Chức năng tìm kiếm sản phẩm

Thông tin sản phẩm Thêm Mới Sản Phẩm

Tìm kiếm theo mã sản phẩm

Nhập từ khóa :

1001

Làm mới

Tìm Kiếm

Sản Phẩm

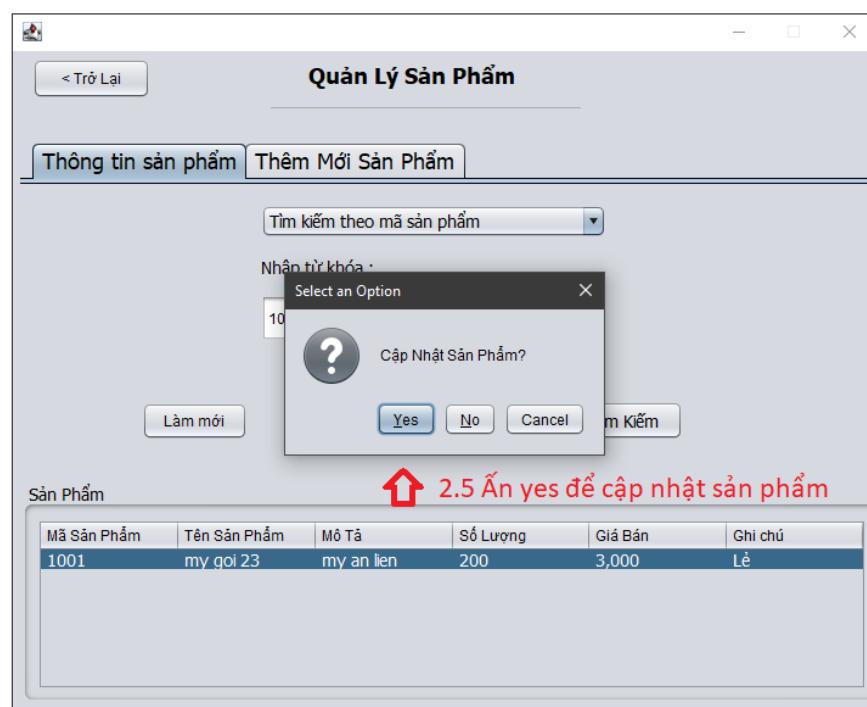
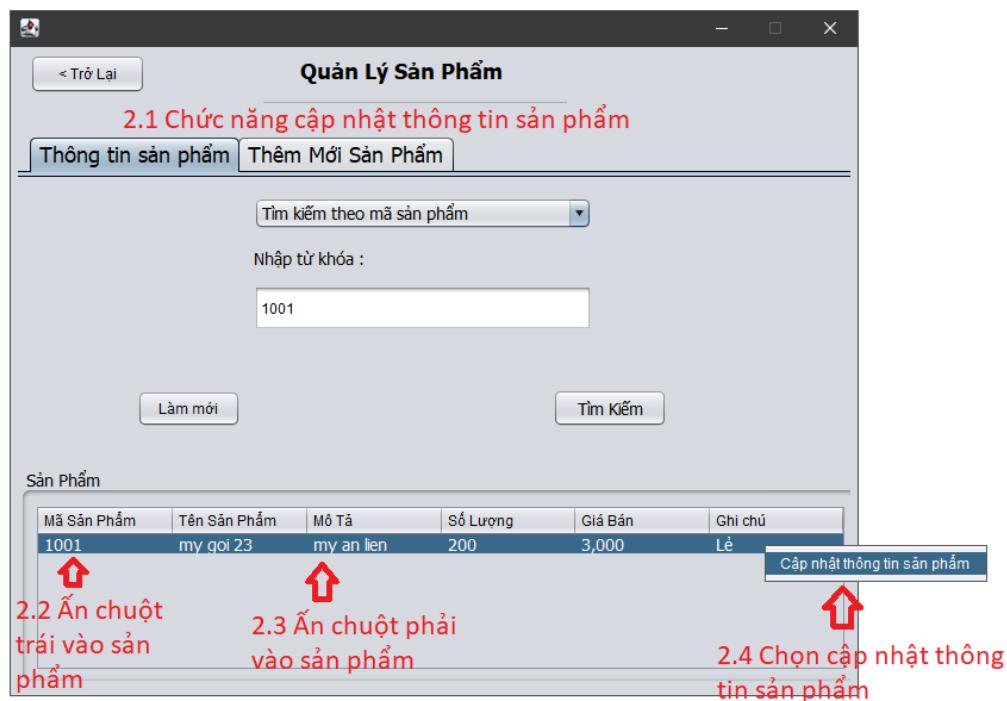
Mã Sản Phẩm	Tên Sản Phẩm	Mô Tả	Số Lượng	Giá Bán	Ghi chú
1001	mỳ gói 23	mỳ ăn liền	200	3,000	Lé

1.2 Chọn hình thức tìm kiếm

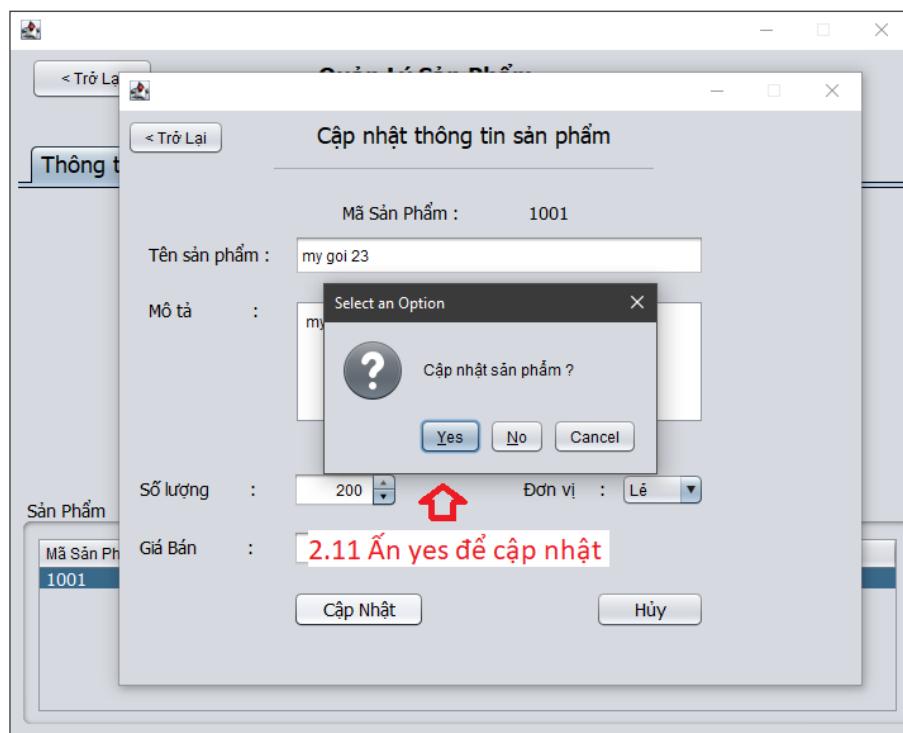
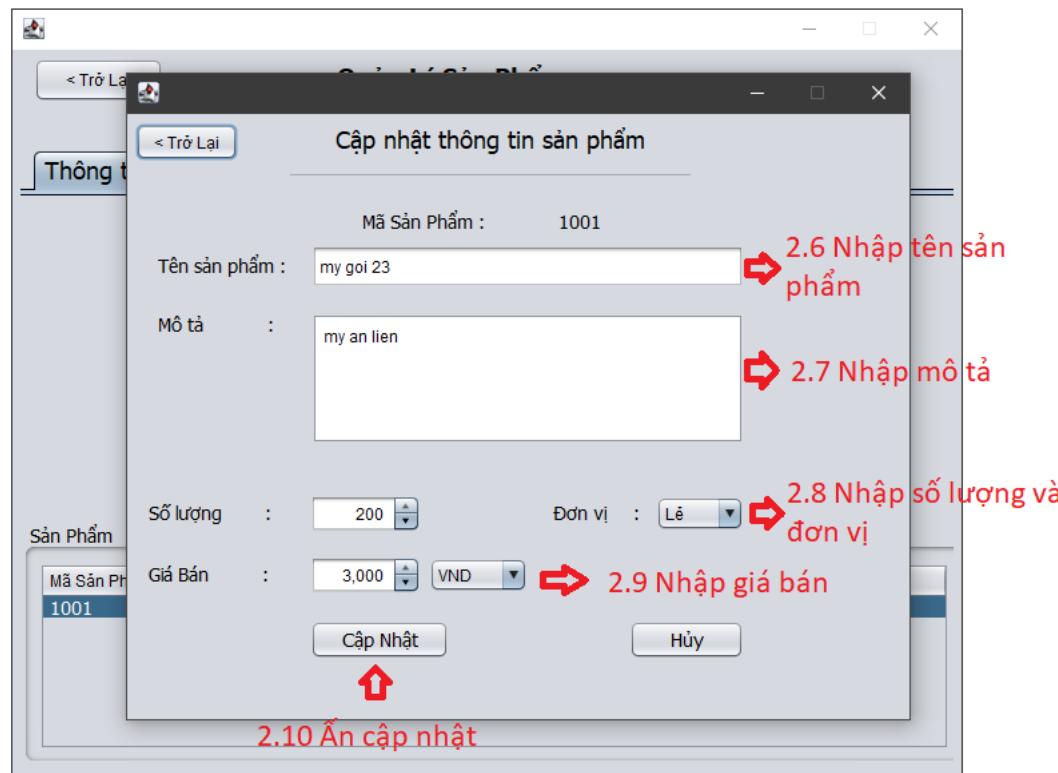
1.3 Nhập từ khóa cần tìm

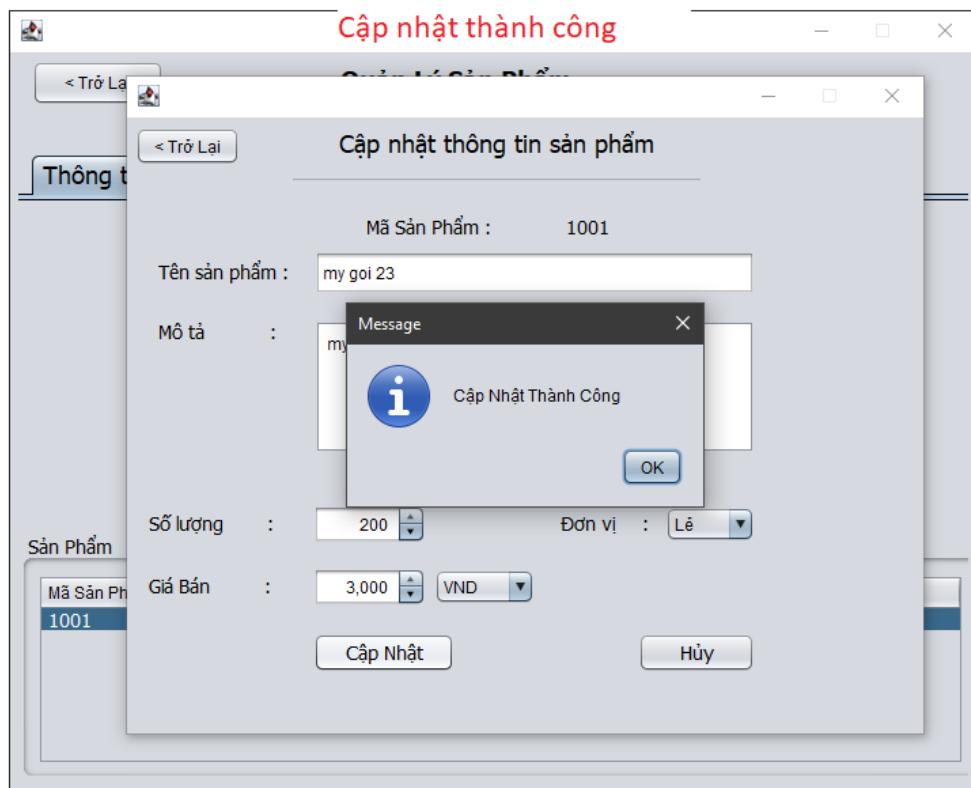
1.4 Tìm kiếm

1.5 Kết quả



Giao diện chỉnh sửa thông tin sản phẩm





3 Chức năng thêm mới sản phẩm

Thêm mới sản phẩm

Mã sản phẩm : 1009

Tên sản phẩm : Chanh muối

Mô tả : Giải khát

Số Lượng : 1,000 Đơn vị : Lê

Giá bán : 10,000 VND

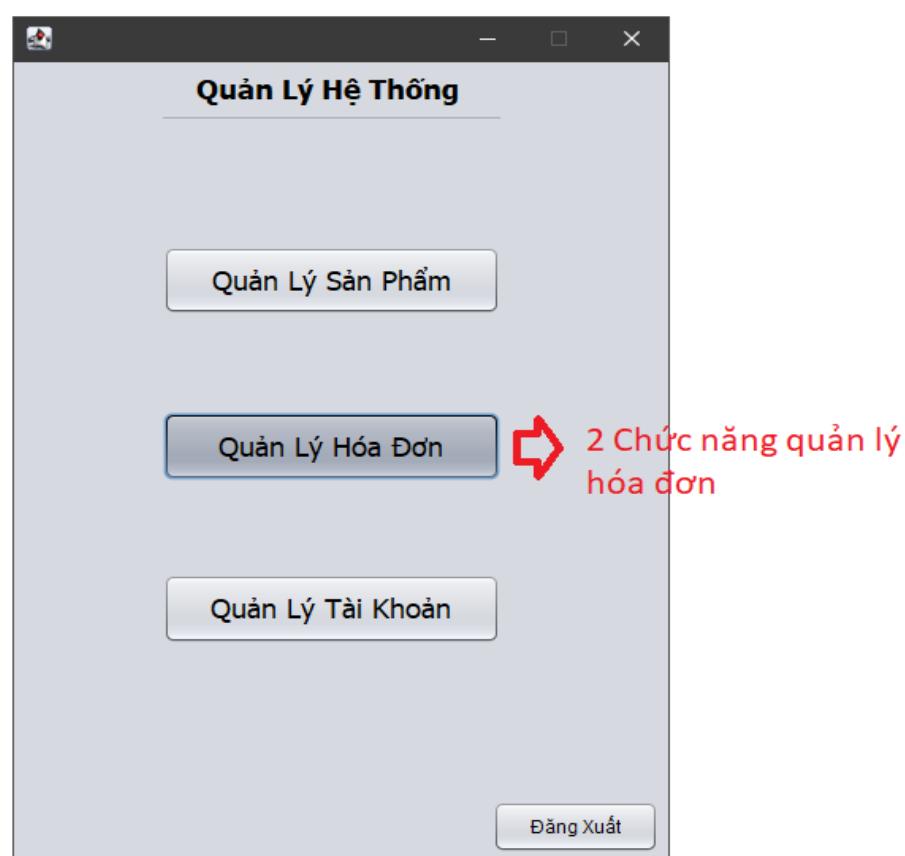
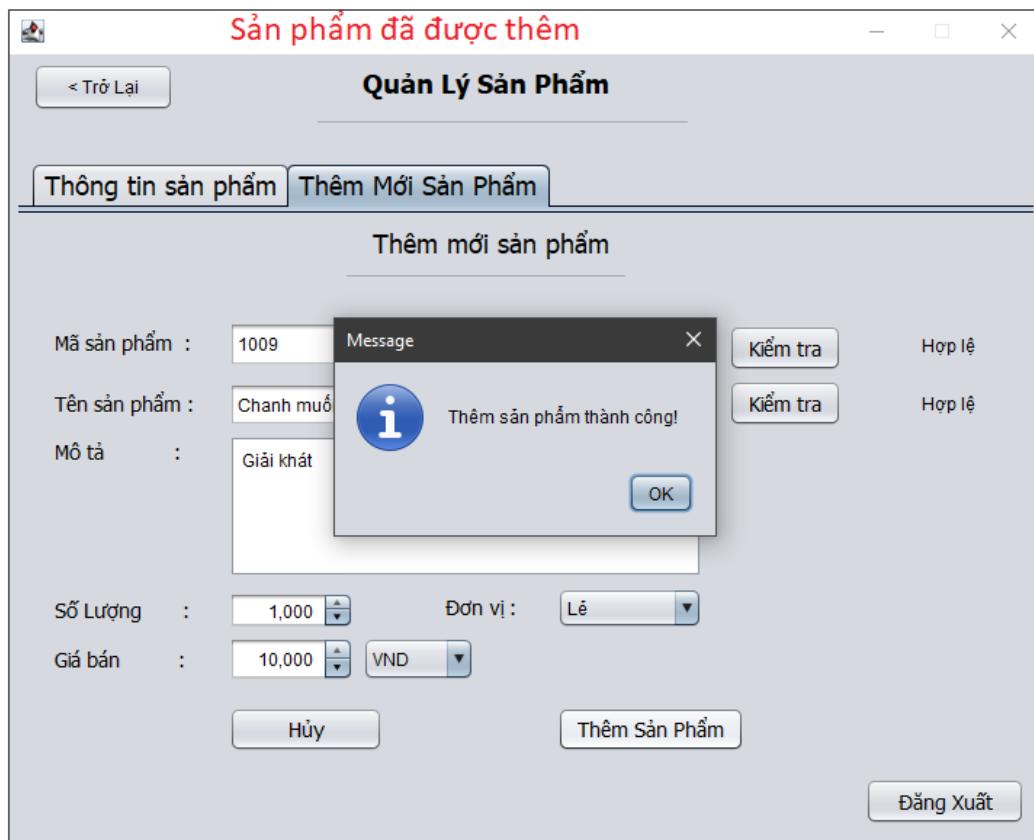
Kiểm tra Hợp lệ

Hợp lệ

3.1 Nhập thông tin sản phẩm

3.2 Kiểm tra sản phẩm đã tồn tại trong hệ thống chưa

3.4 Thêm sản phẩm



[← Trở Về](#) 1 Giao diện thống kê hóa đơn **Thống kê doanh thu**

2.1 Ấn tìm kiếm

Thống kê từ ngày : Ngày: 1 Tháng: 01 Năm: 2020

Đến ngày : Ngày: 16 Tháng: 09 Năm: 2020 Hiện Tại

Tìm kiếm **Làm mới** **In Thống Kê**

1.1 Chọn ngày thống kê

Mã Hóa Đơn	Nhân Viên Bán Hàng	Ngày Bán	Giờ Bán	Tổng Tiền Hóa Đơn
------------	--------------------	----------	---------	-------------------

Tổng Số Hóa Đơn : **Tổng Tiền Hóa Đơn :**

Thống kê doanh thu

Thống kê từ ngày : Ngày: Tháng: Năm:

Đến ngày : Ngày: Tháng: Năm:

Hiện Tại

 1.3 Ấn in thống kê

Mã Hóa Đơn	Nhân Viên Bán Hàng	Ngày Bán	Giờ Bán	Tổng Tiền Hóa Đơn
1	thang	15-09-2020	23:44:22	2,000
2	huan	15-09-2020	23:44:22	2,000
3	huan	15-09-2020	23:44:22	6,000
4	huan	15-09-2020	23:44:22	3,000
5	huan	15-09-2020	23:44:22	6,000
6	huan	15-09-2020	23:44:22	3,000
7	huan	15-09-2020	23:44:22	6,000
8	huan	15-09-2020	23:44:22	6,000

 Hiện thị kết quả tìm kiếm

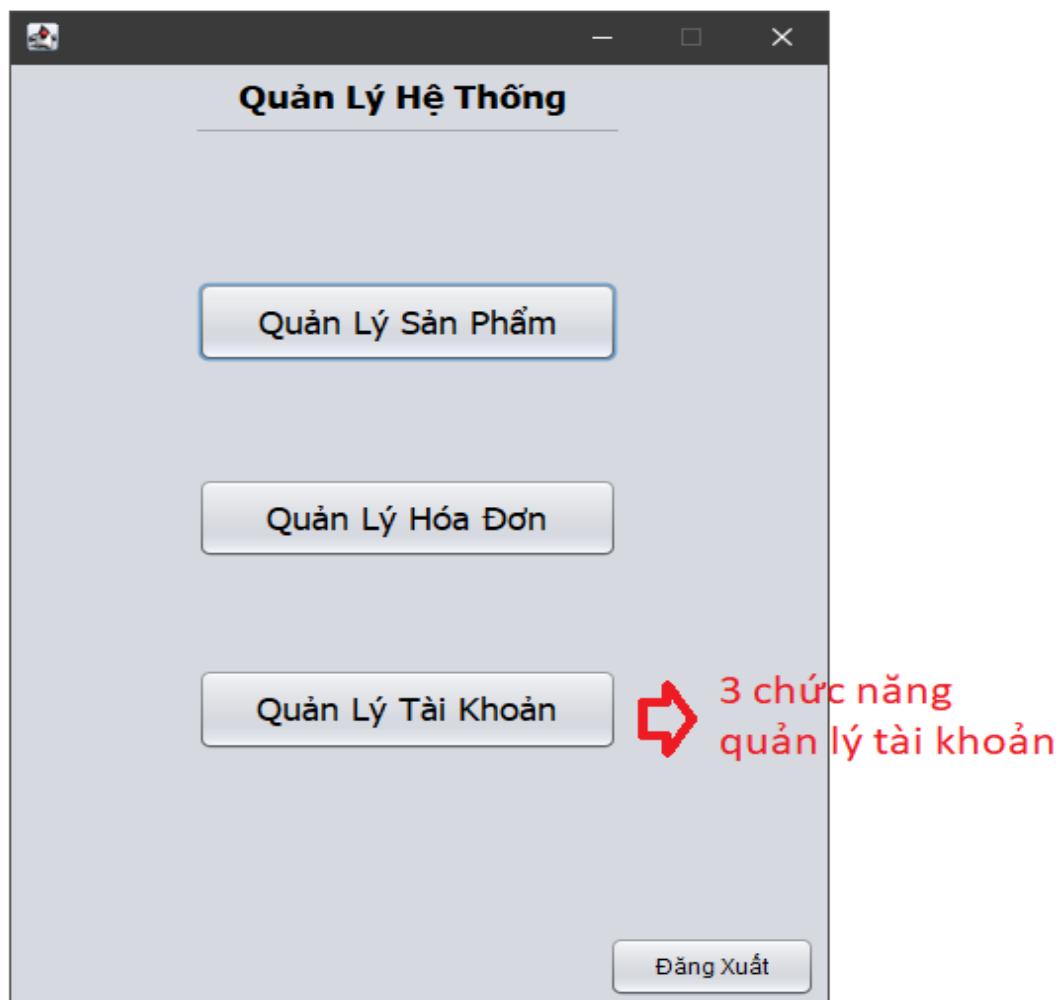
Tổng Số Hóa Đơn : 8 Tổng Tiền Hóa Đơn : 34,000

Bảng thống kê hóa đơn

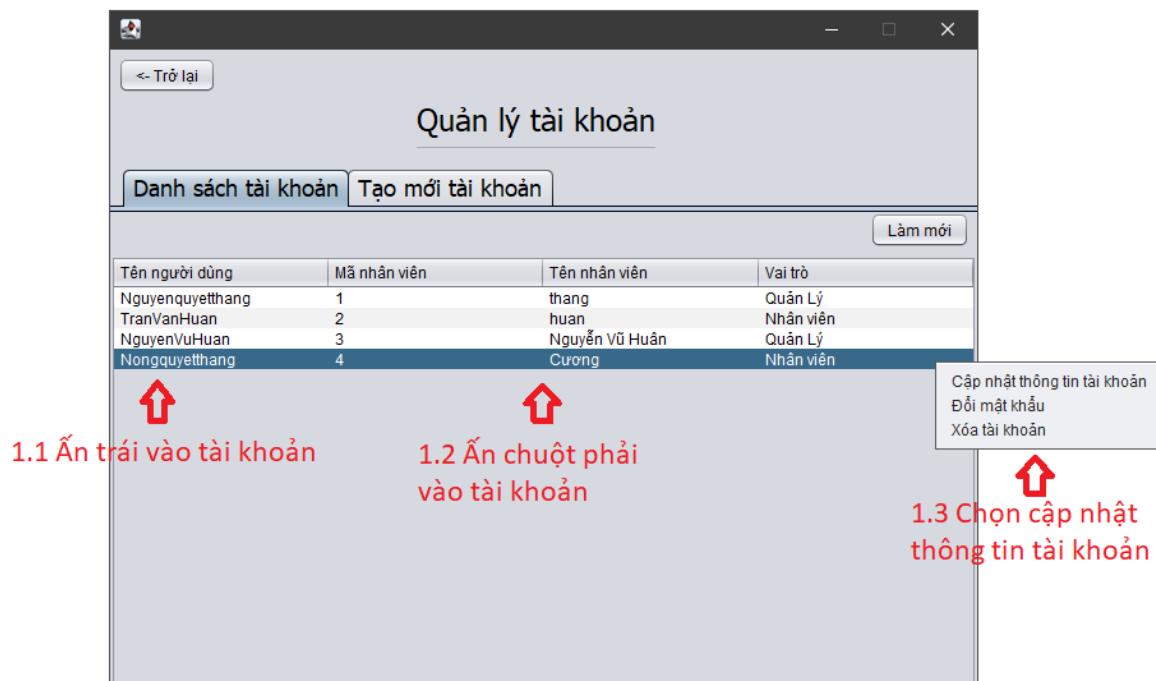
THÔNG KÊ HÓA ĐƠN BÁN HÀNG

Mã hóa đơn	Nhân viên bán hàng	Ngày bán	Giờ bán	Tổng tiền hóa đơn
1	thang	15-09-2020	23:44:22	2,000
2	huan	15-09-2020	23:44:22	2,000
3	huan	15-09-2020	23:44:22	6,000
4	huan	15-09-2020	23:44:22	3,000
5	huan	15-09-2020	23:44:22	6,000
6	huan	15-09-2020	23:44:22	3,000
7	huan	15-09-2020	23:44:22	6,000
8	huan	15-09-2020	23:44:22	6,000

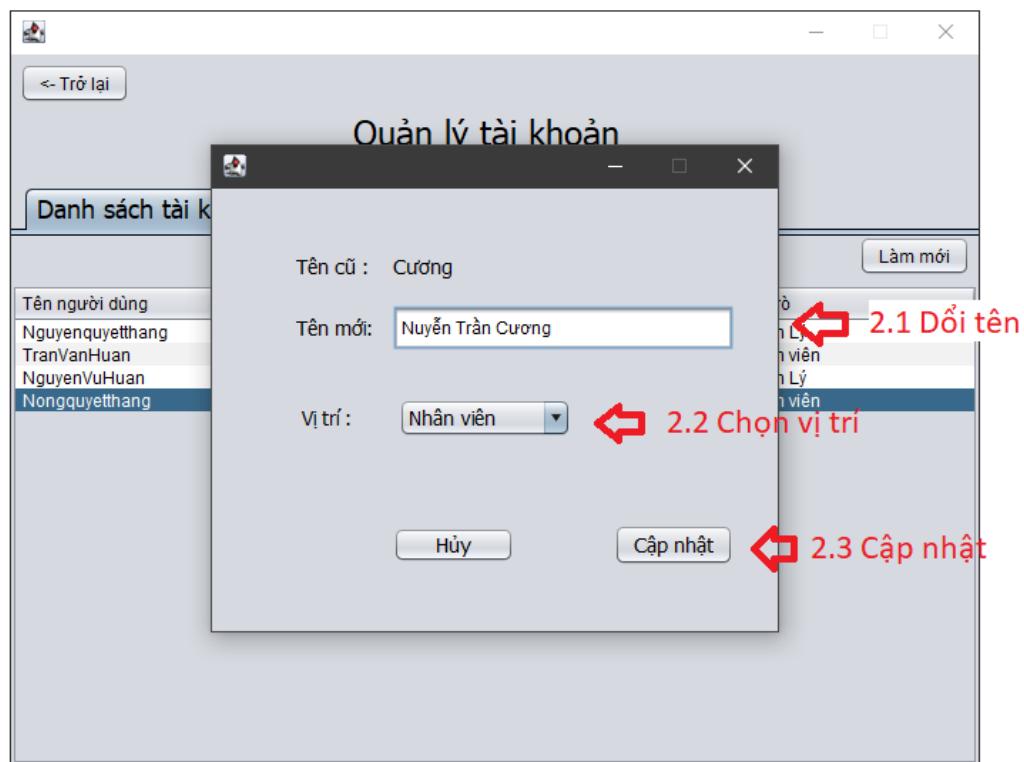
Người lập biểu
(Ký, ghi rõ họ tên) Kế toán trưởng
(Ký, ghi rõ họ tên) Người đại diện theo pháp luật
(Ký, ghi rõ họ tên, đóng dấu)

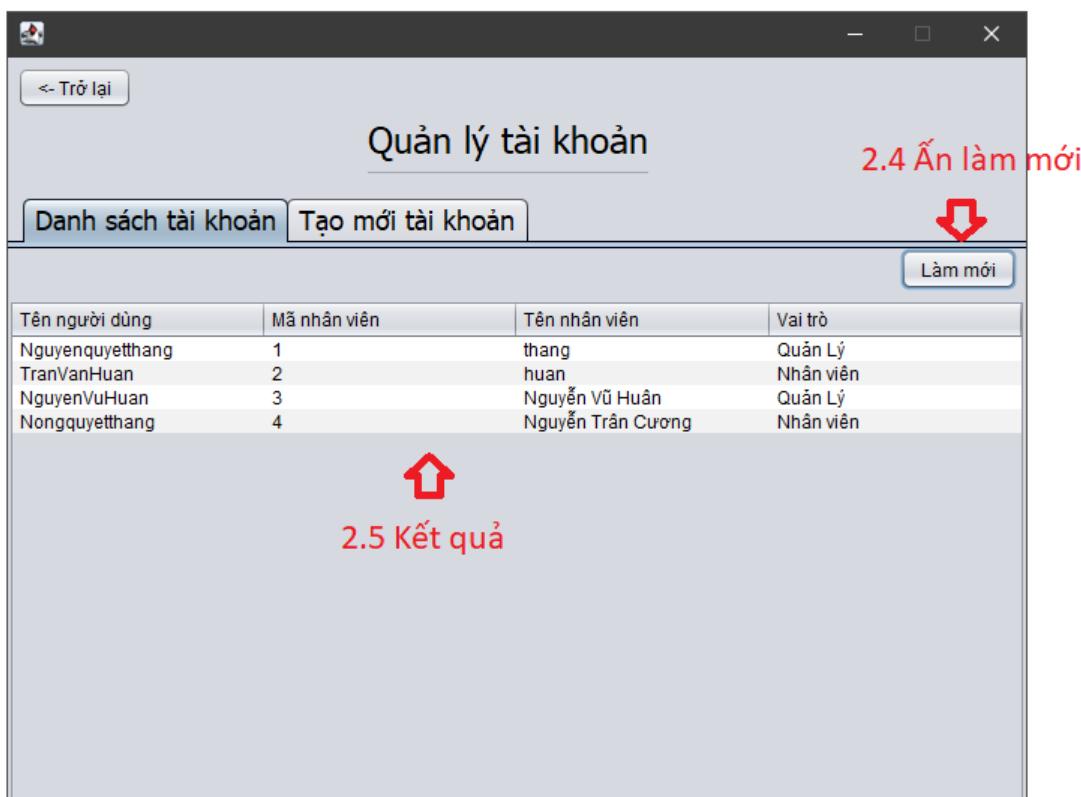
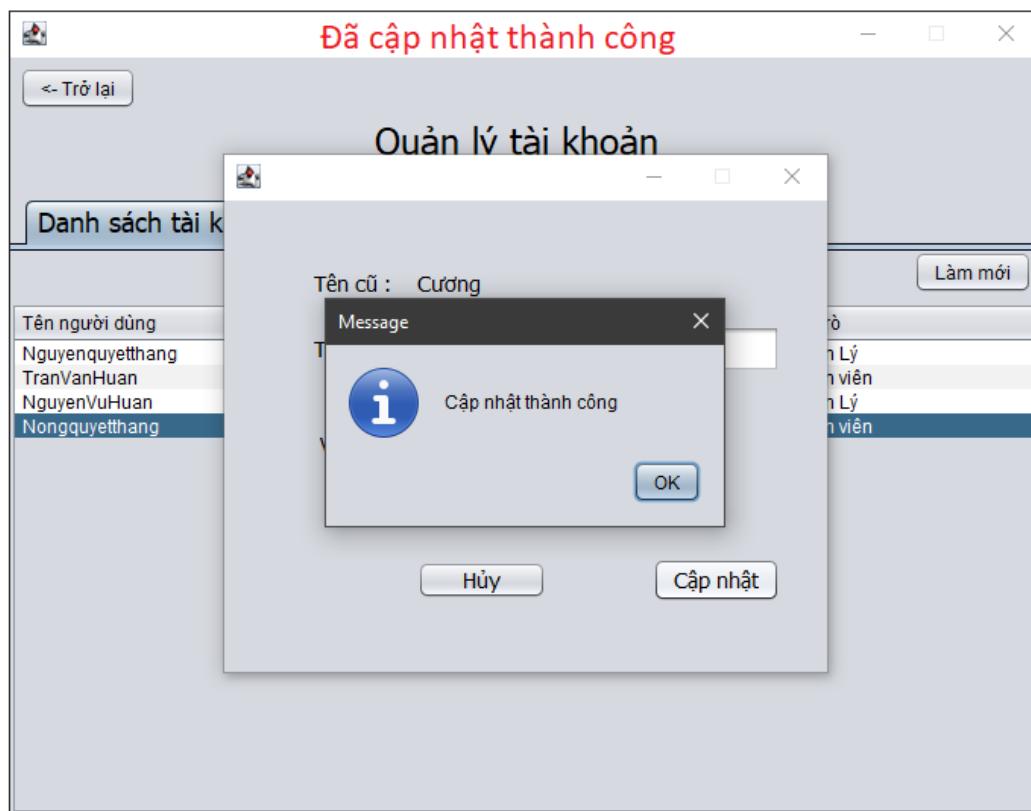


1 Chức năng chỉnh sửa thông tin tài khoản

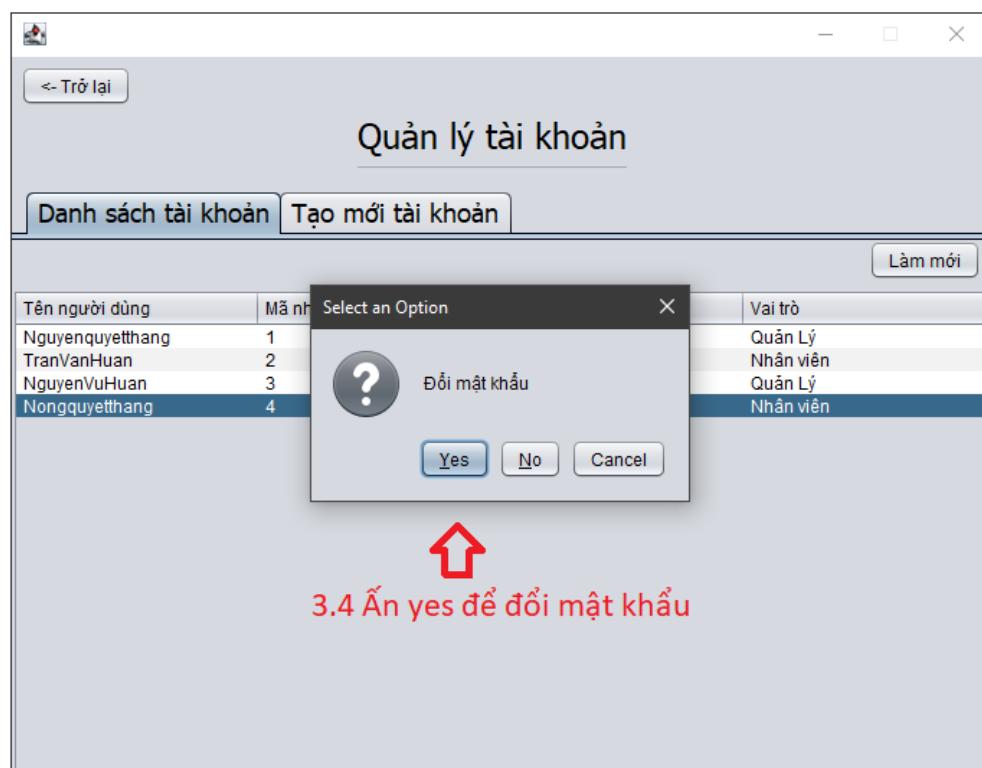
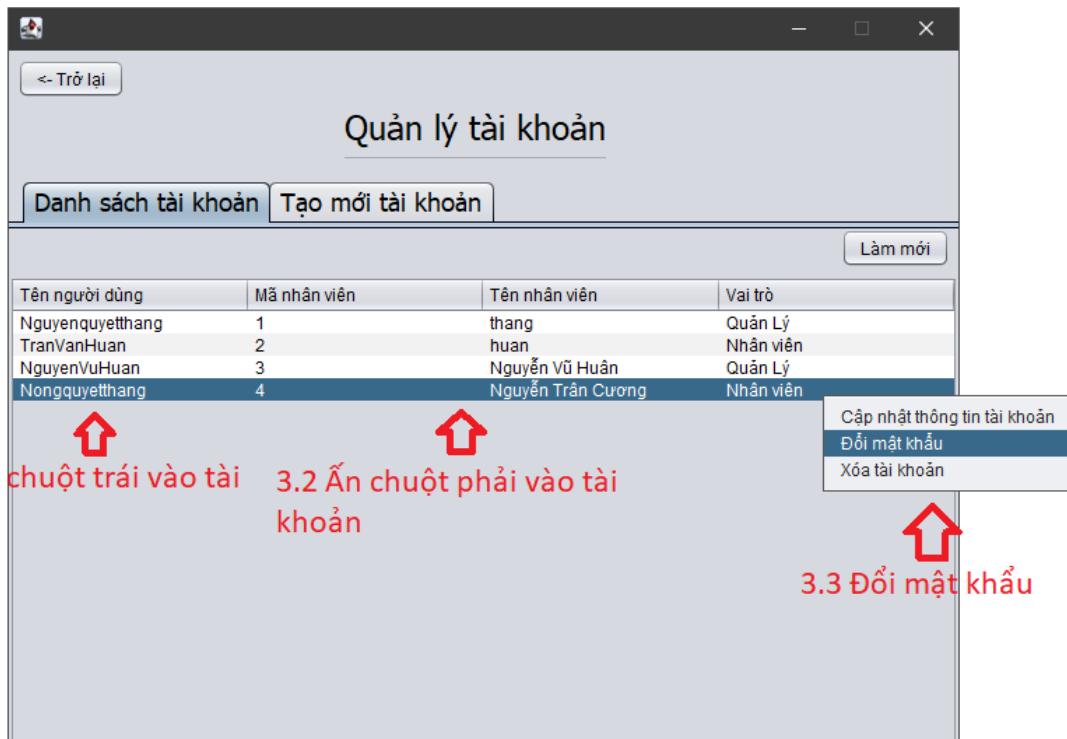


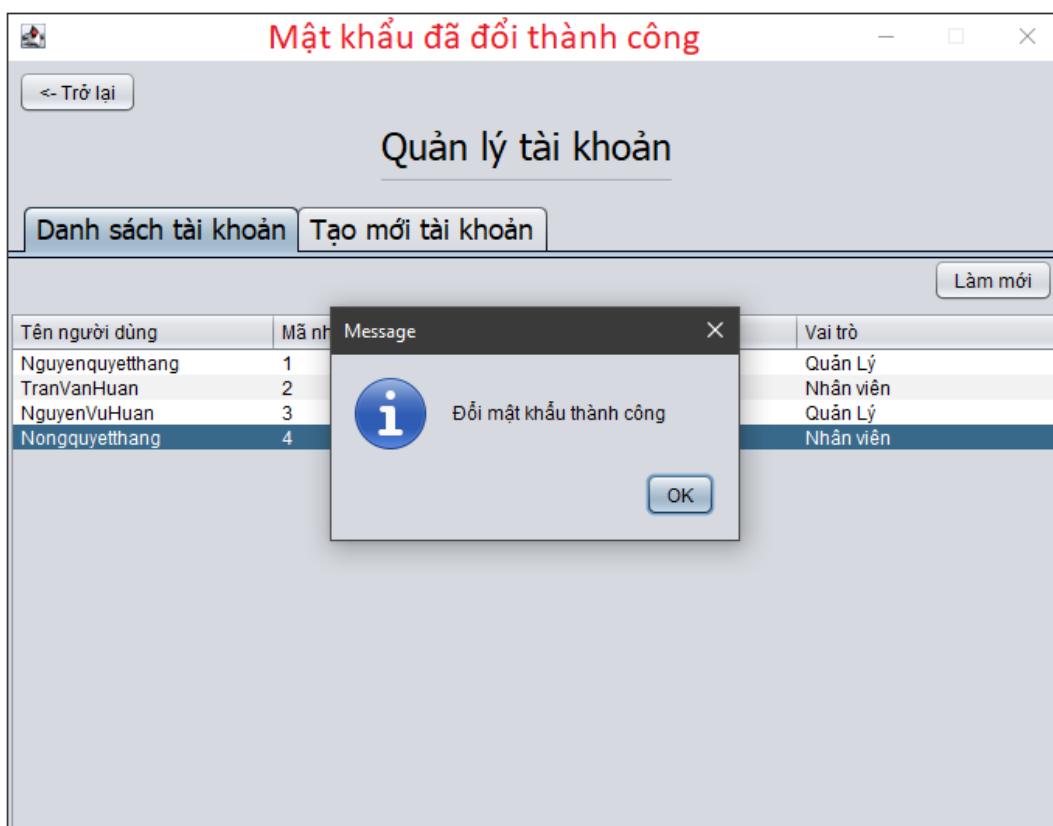
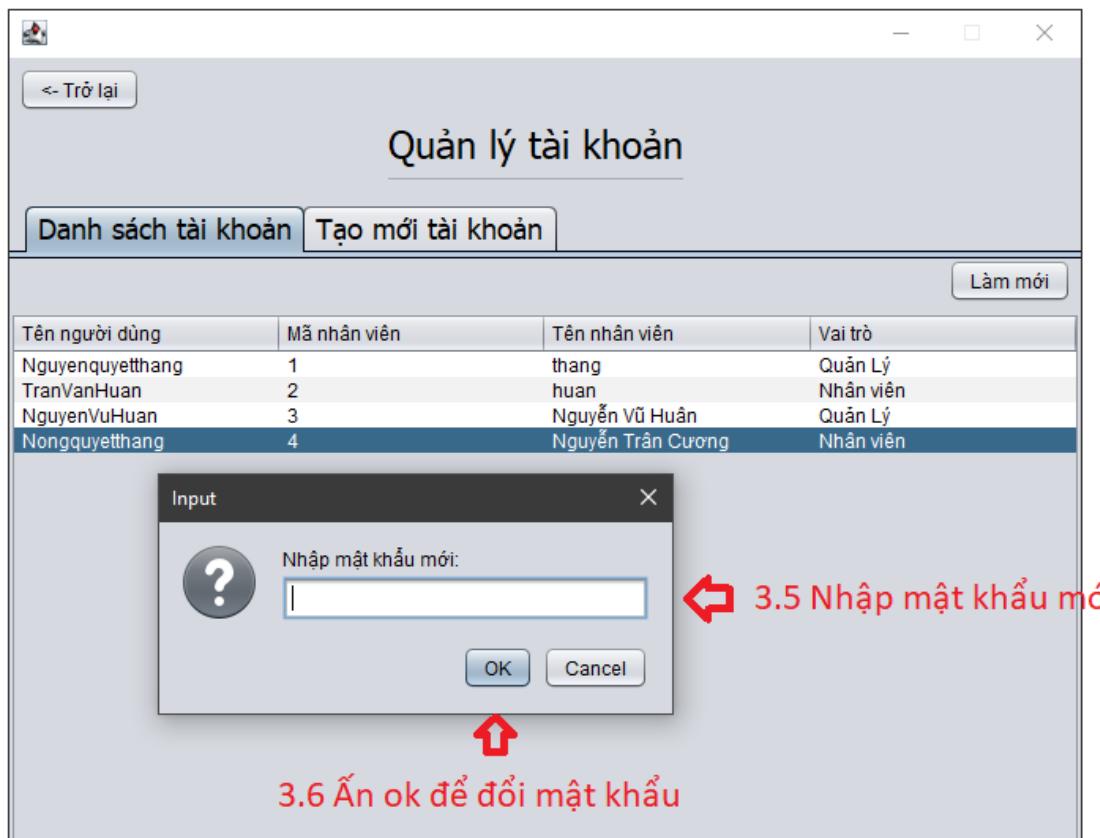
Giao diện cập nhật thông tin tài khoản





3 Chức năng đổi mật khẩu





4 Chức năng tạo tài khoản

Quản lý tài khoản

Danh sách tài khoản **Tạo mới tài khoản**

Nhập tên đăng nhập : Kiểm tra Hợp lệ ↗ 4.2 kiểm tra

Tên của bạn :

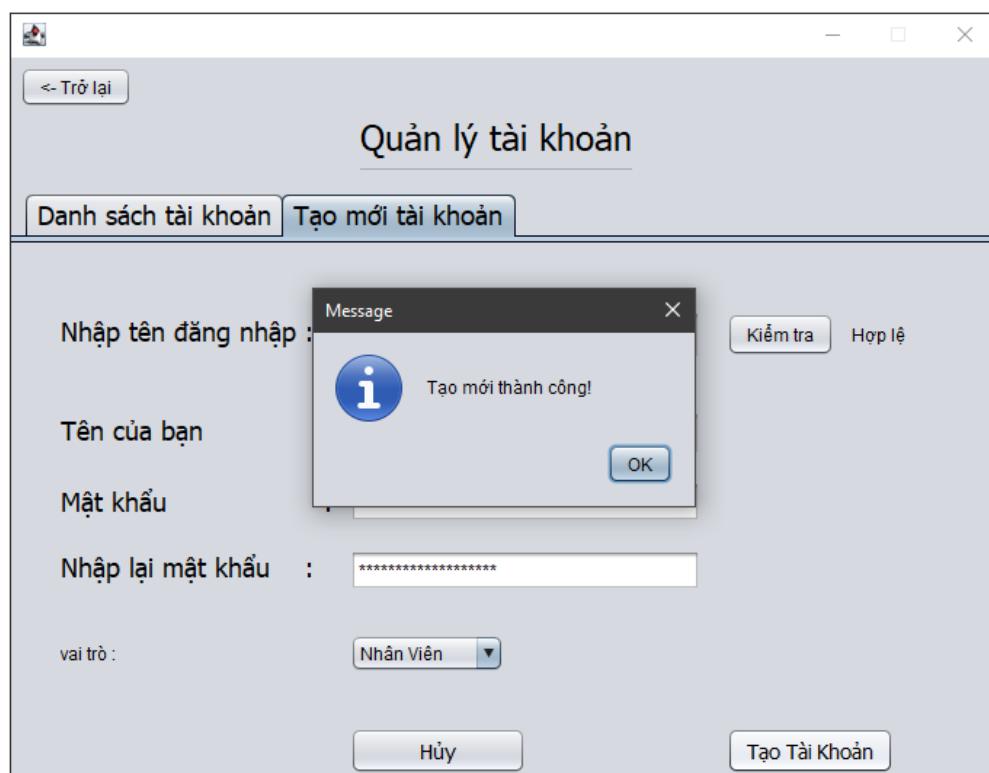
Mật khẩu :

Nhập lại mật khẩu :

vai trò :

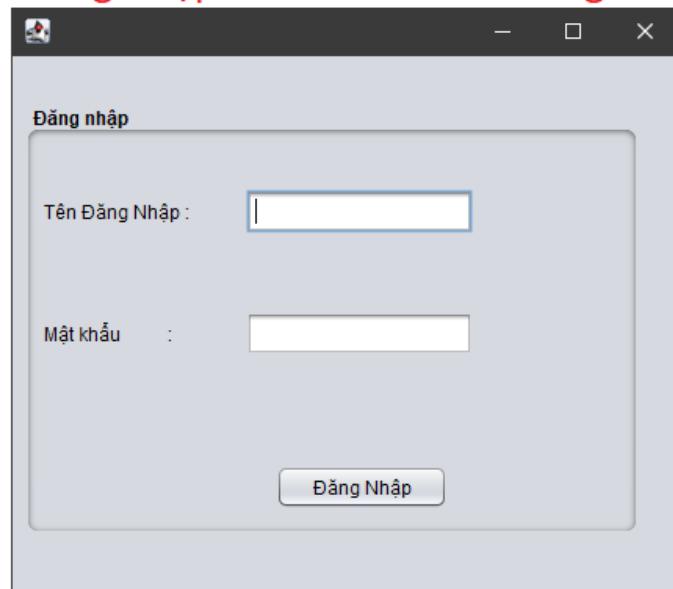
Hủy Tạo Tài Khoản ↗ 4.3 tạo tài khoản

Tài khoản đã tạo thành công



2. Cashier

Đăng nhập tài khoản của thu ngân



Giao diện tạo hóa đơn

Tạo mới hóa đơn

Đăng xuất

1 Chọn hình thức tìm kiếm

2 Nhập từ khóa cần tìm kiếm

3 Ấn tìm kiếm

Hình thức tìm kiếm

Nhập từ khóa :

Kết quả :

Mã sản phẩm	Tên sản phẩm	SL	Giá

Thông tin hóa đơn

Số hóa đơn : Ngày lập : Tên Nhân Viên :

Cơ sở kinh doanh : VTC academy HN Địa chỉ : Hà Nội

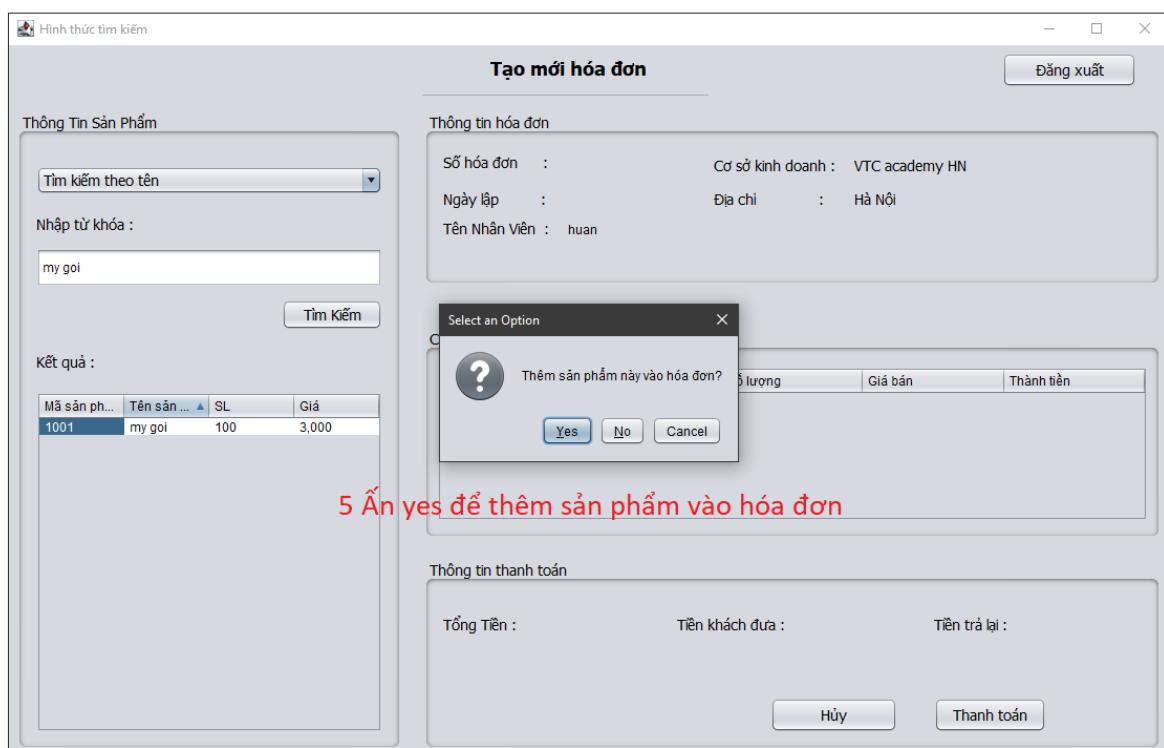
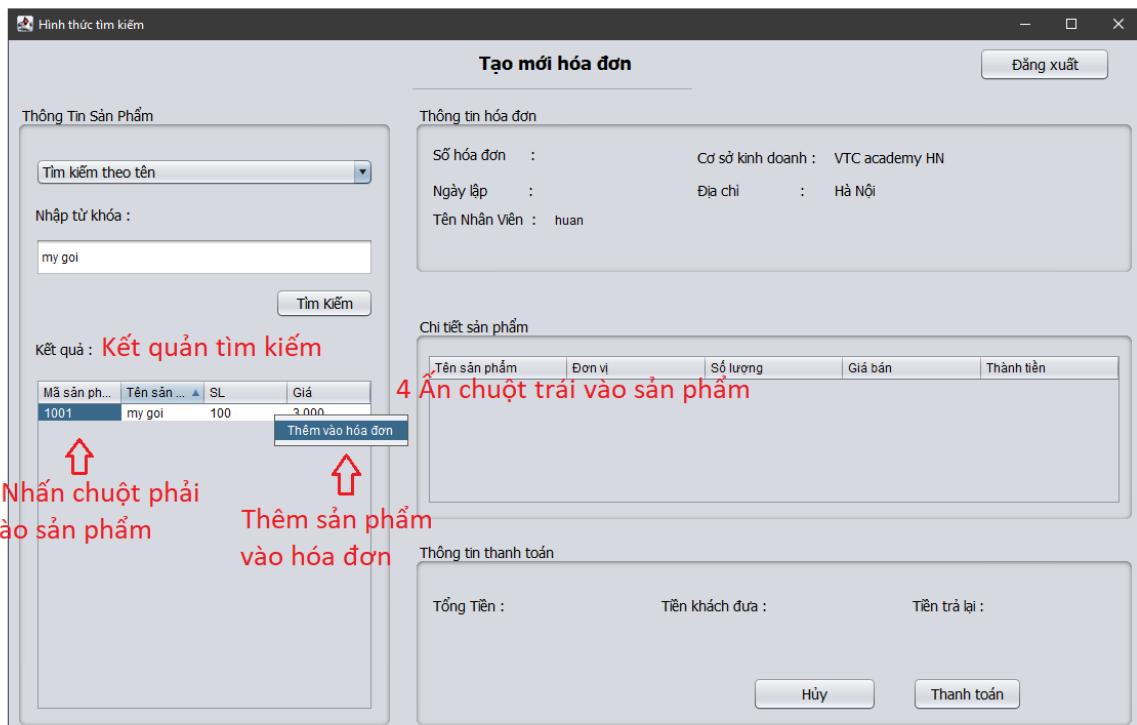
Chi tiết sản phẩm

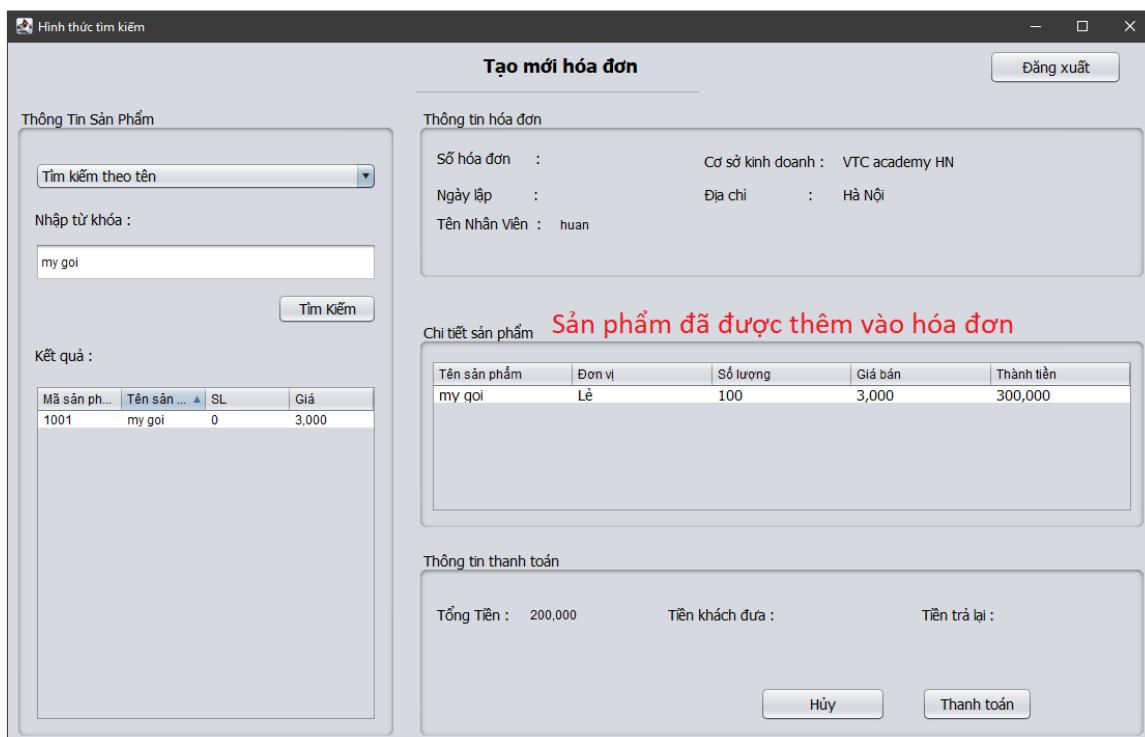
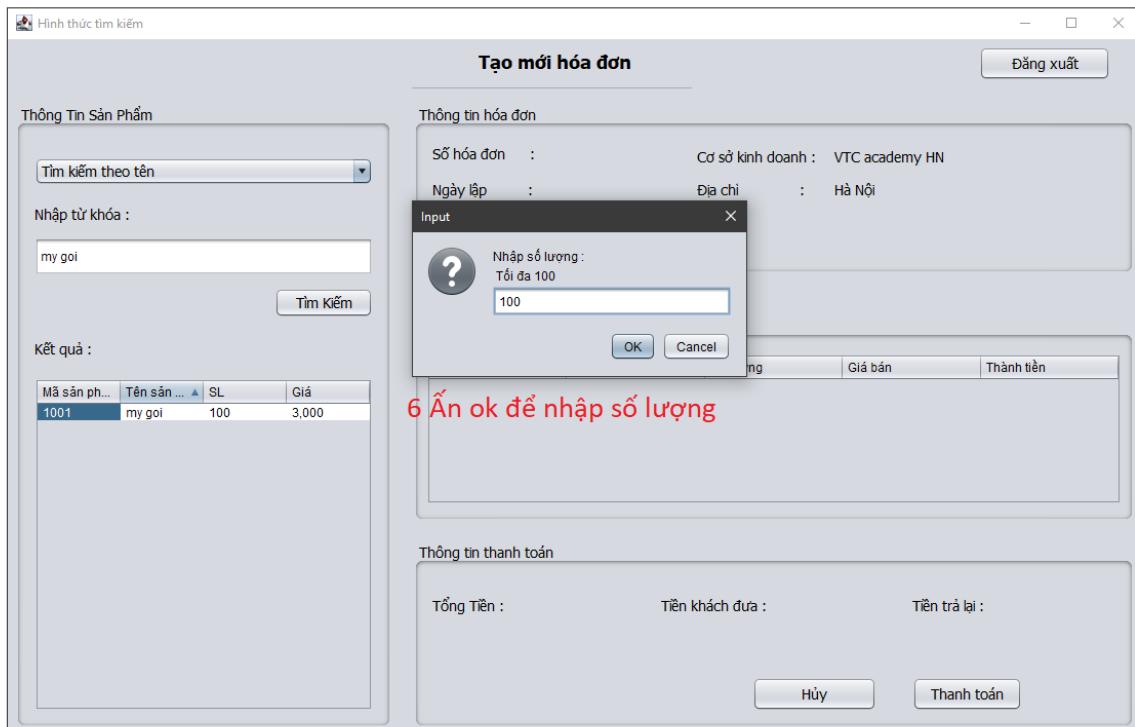
Tên sản phẩm	Đơn vị	Số lượng	Giá bán	Thành tiền

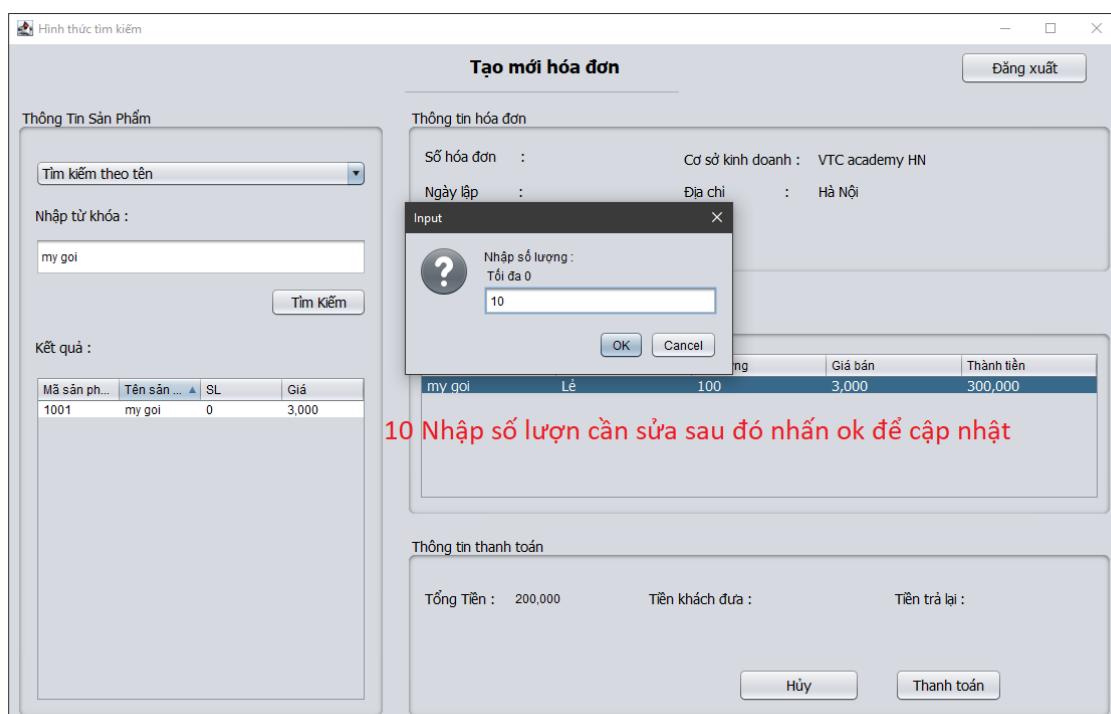
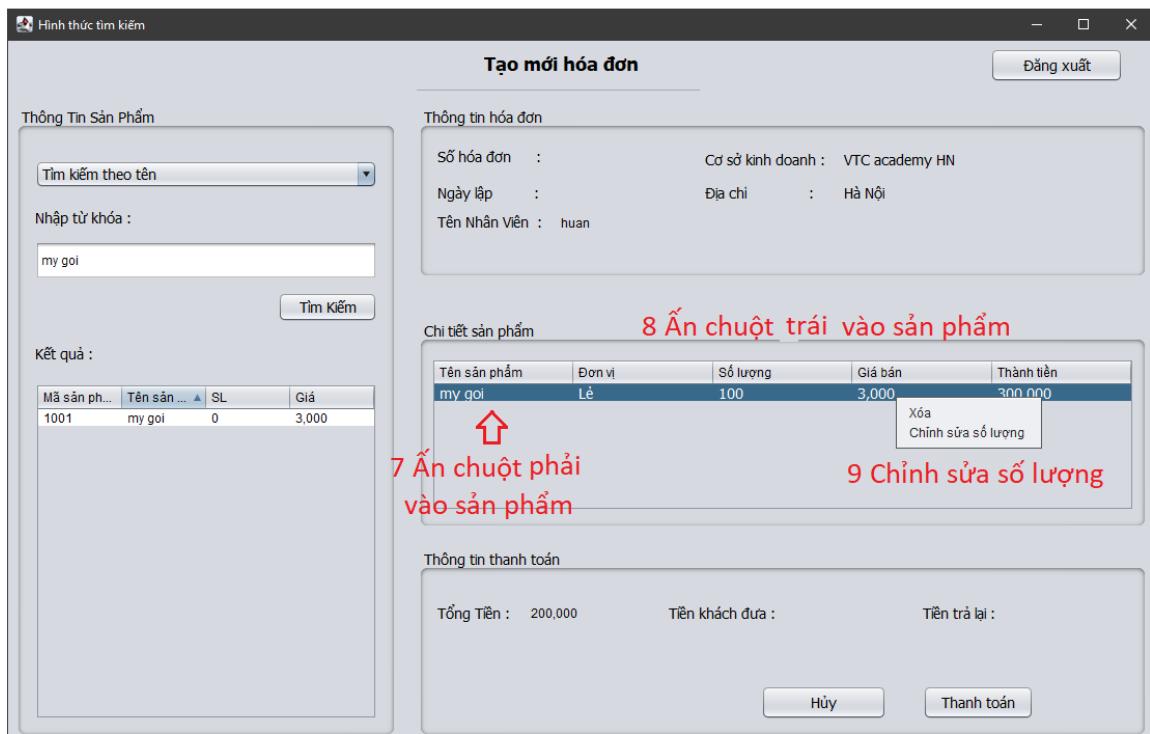
Thông tin thanh toán

Tổng Tiền : Tiền khách đưa : Tiền trả lại :

Hủy Thanh toán







Hình thức tìm kiếm

Tạo mới hóa đơn

Đăng xuất

Thông tin Sản Phẩm

Tìm kiếm theo tên

Nhập từ khóa :

Tìm Kiếm

Kết quả :

Mã sản phẩm	Tên sản phẩm	SL	Giá
1001	my goi	90	3.000

Thông tin hóa đơn

Số hóa đơn : Cơ sở kinh doanh : VTC academy HN
 Ngày lập : Địa chỉ : Hà Nội
 Tên Nhân Viên : huan

Chi tiết sản phẩm

Tên sản phẩm	Đơn vị	Số lượng	Giá bán	Thành tiền
my goi	Lẻ	10	3,000	30,000

Sản phẩm đã được chỉnh sửa

Thông tin thanh toán

Tổng Tiền : 20,000 Tiền khách đưa : Tiền trả lại :

Hủy Thanh toán

Hình thức tìm kiếm

Tạo mới hóa đơn

Đăng xuất

Thông tin Sản Phẩm

Tìm kiếm theo mã sản phẩm

Nhập từ khóa :

Tìm Kiếm

Kết quả :

Mã sản phẩm	Tên sản phẩm	SL	Giá
1002	my goi 2	179	3.000

Thông tin hóa đơn

Số hóa đơn : Cơ sở kinh doanh : VTC academy HN
 Ngày lập : Địa chỉ : Hà Nội
 Tên Nhân Viên : huan

Xóa sản phẩm

Chi tiết sản phẩm

Tên sản phẩm	Đơn vị	Số lượng	Giá bán	Thành tiền
my goi	Lẻ	10	3,000	30,000
my goi 2	Lẻ	1	3,000	3,000

11 **Ấn chuột trái sau đó ấn chuột phải vào sản phẩm**

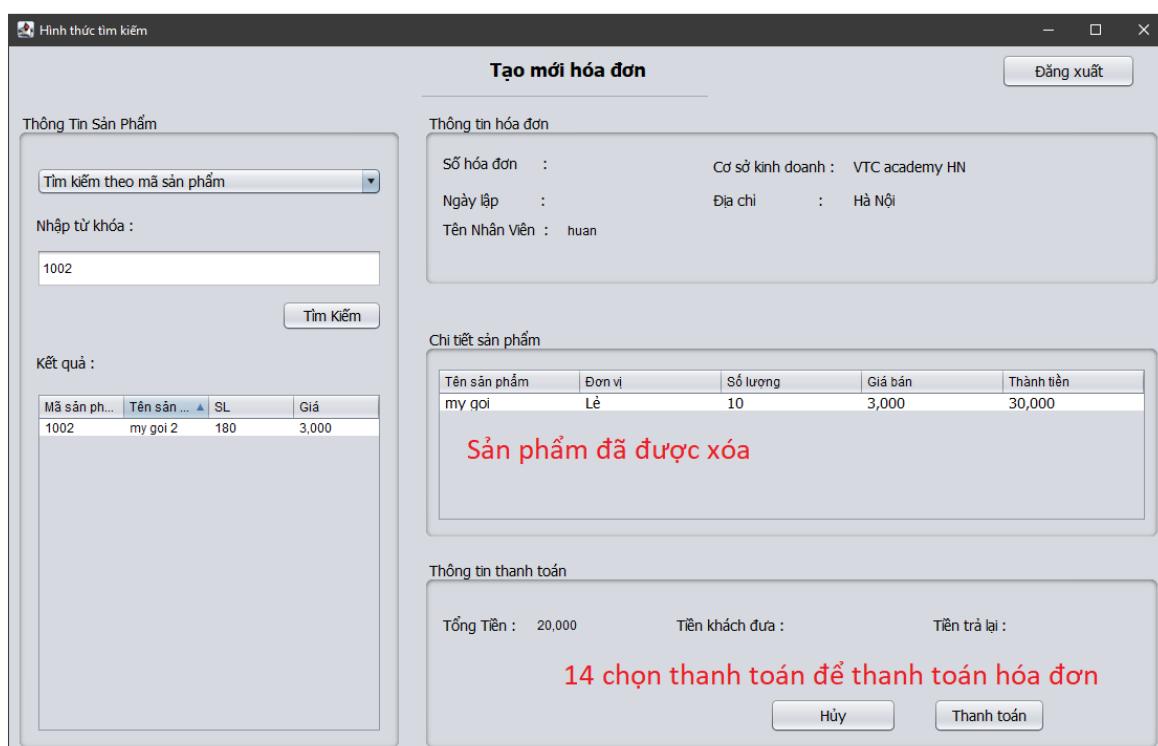
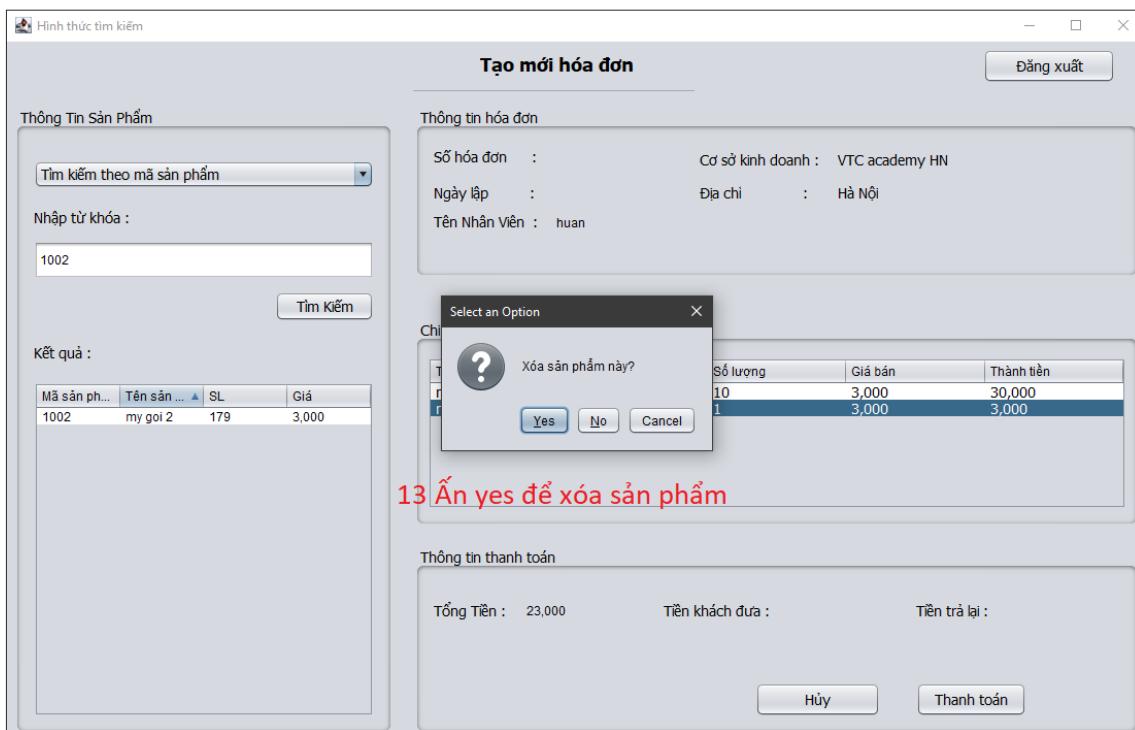
12 **chọn xóa**

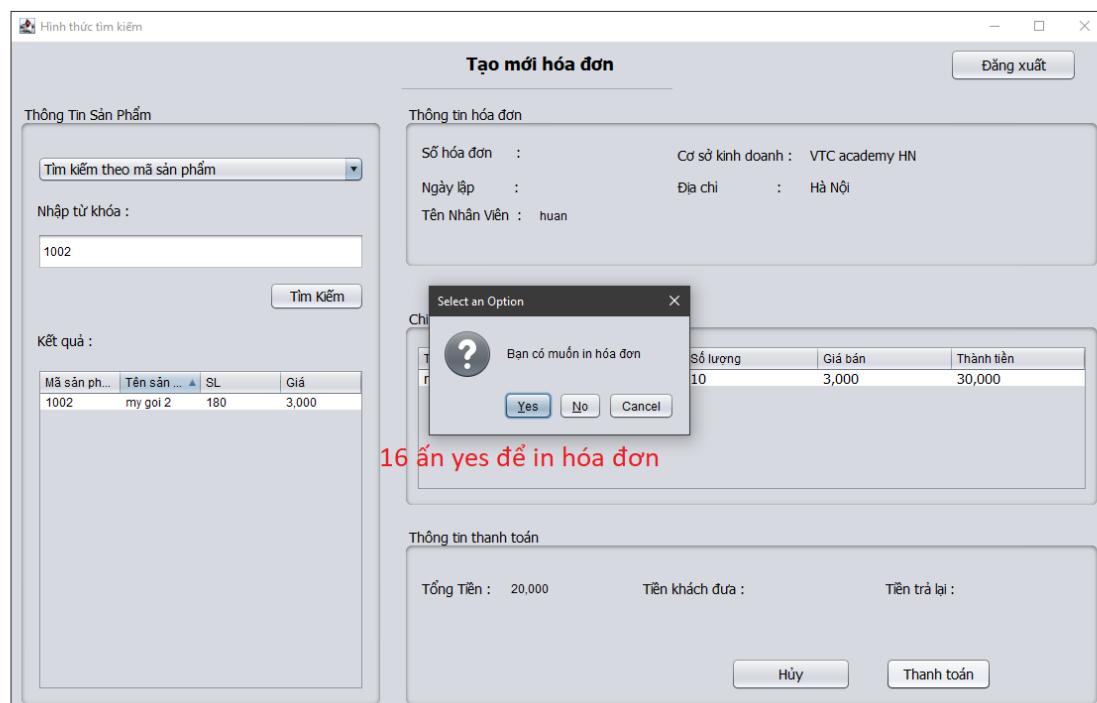
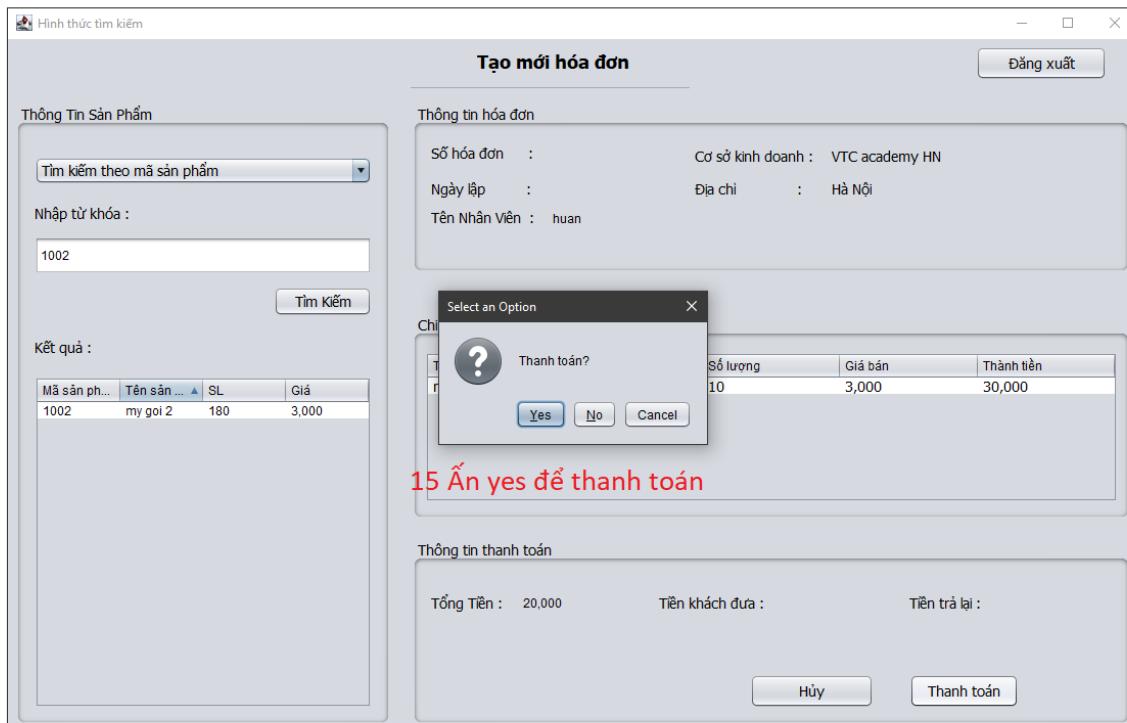
Xóa
Chỉnh sửa số lượng

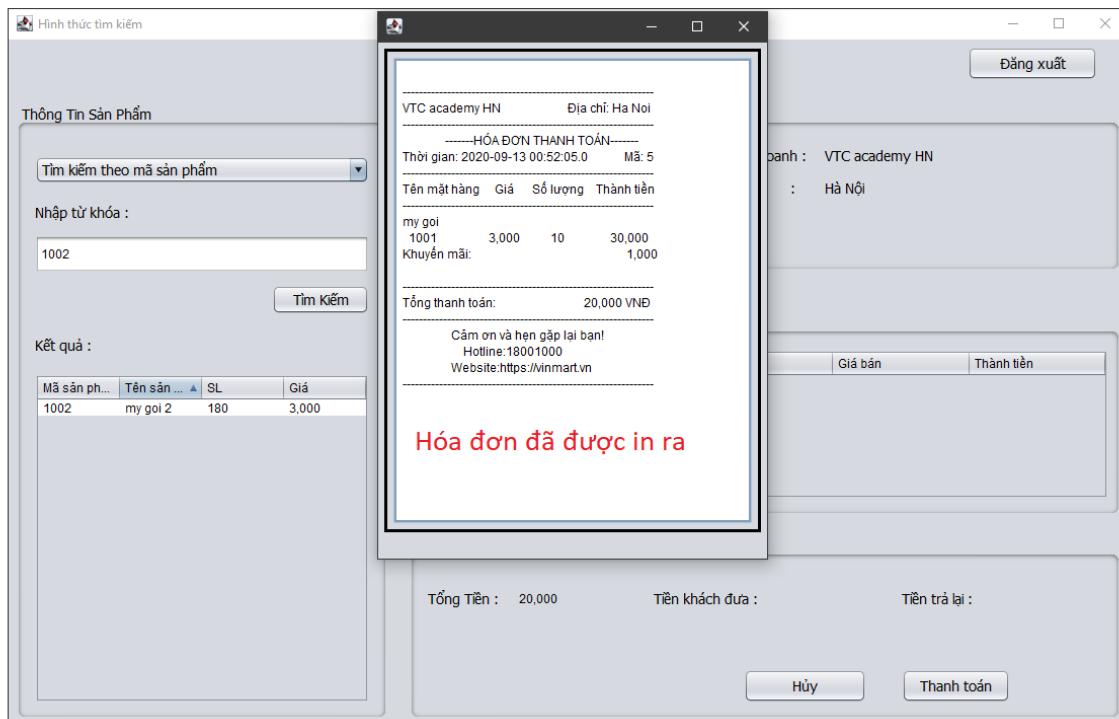
Thông tin thanh toán

Tổng Tiền : 23,000 Tiền khách đưa : Tiền trả lại :

Hủy Thanh toán

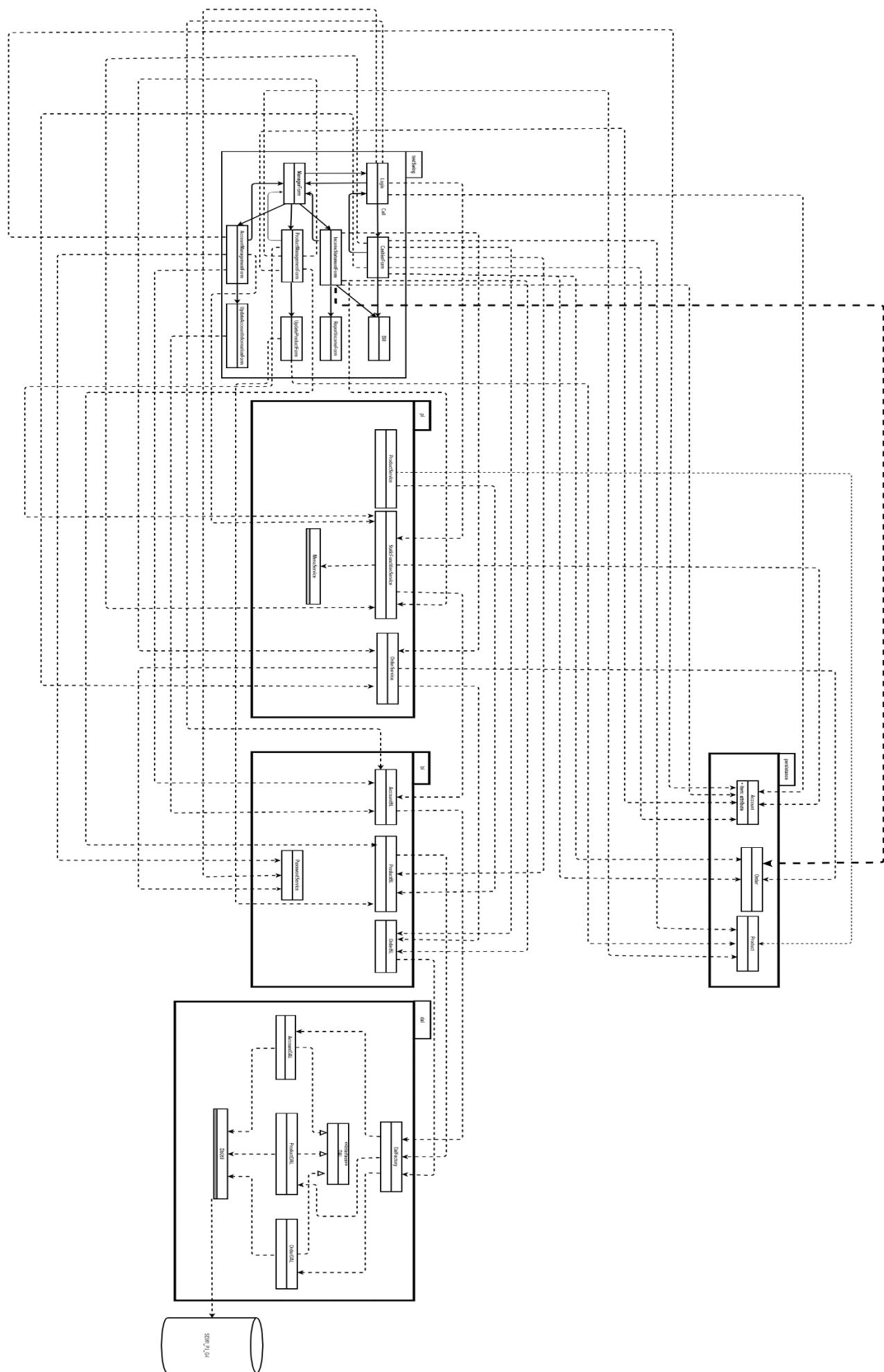


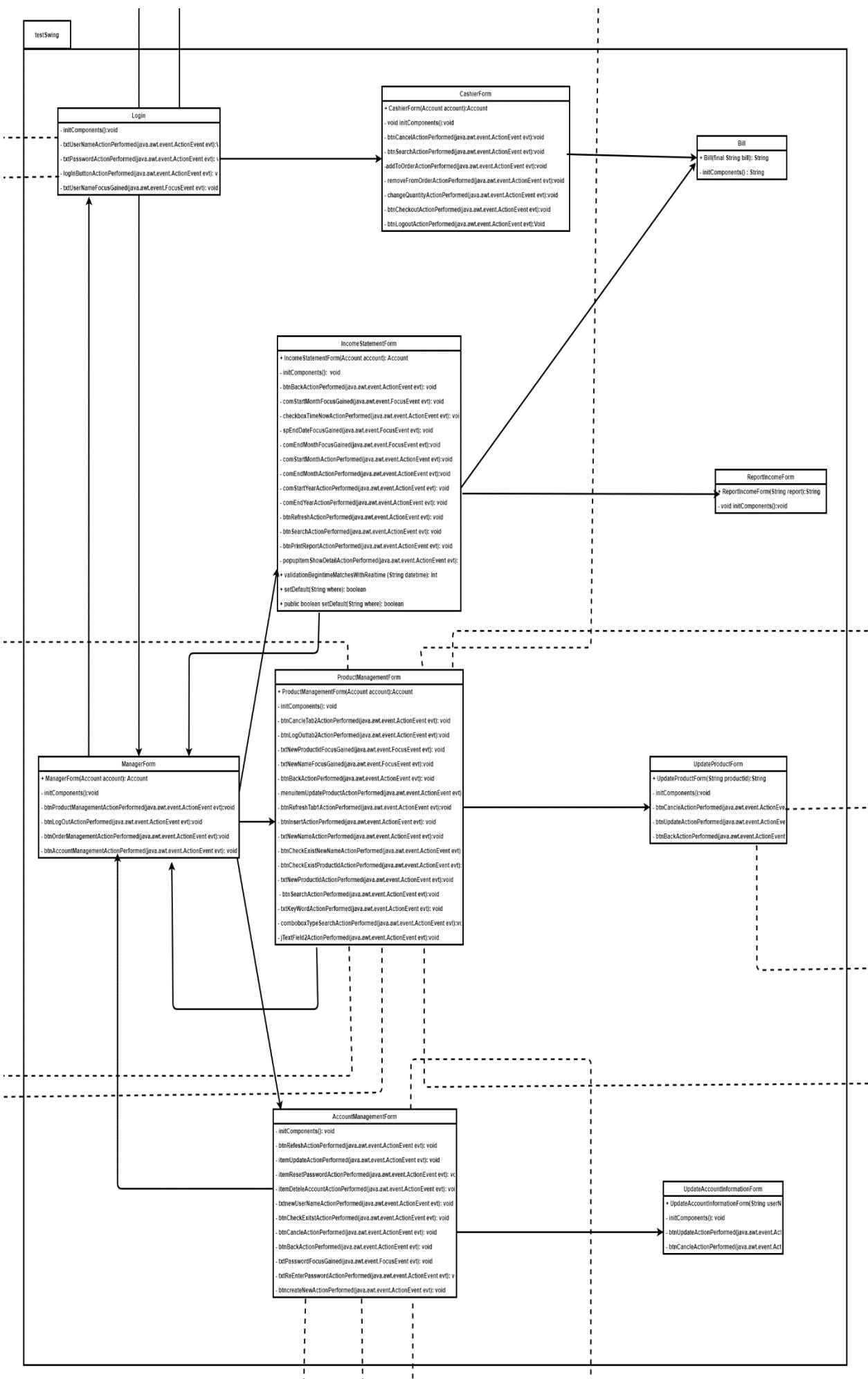


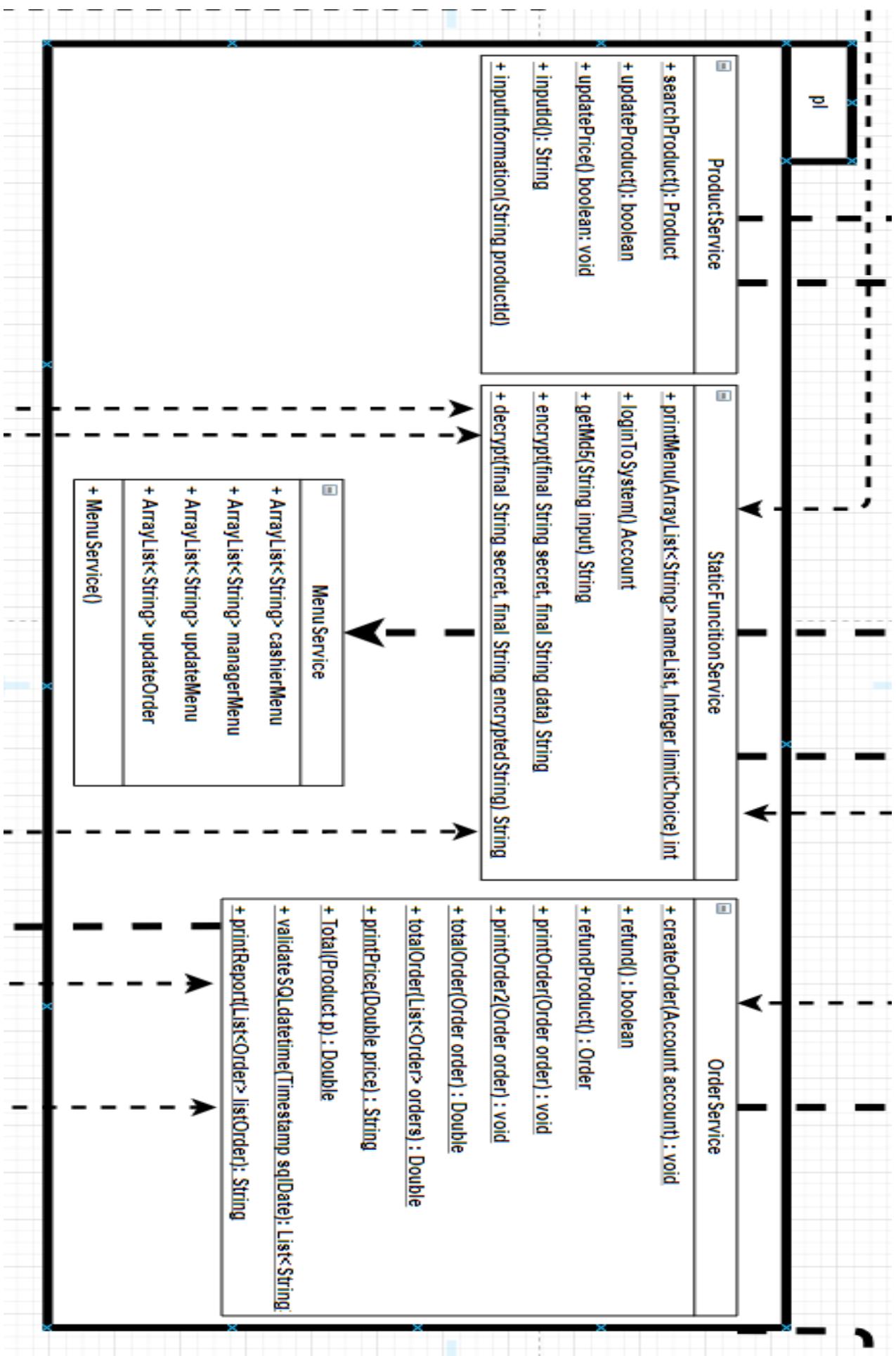


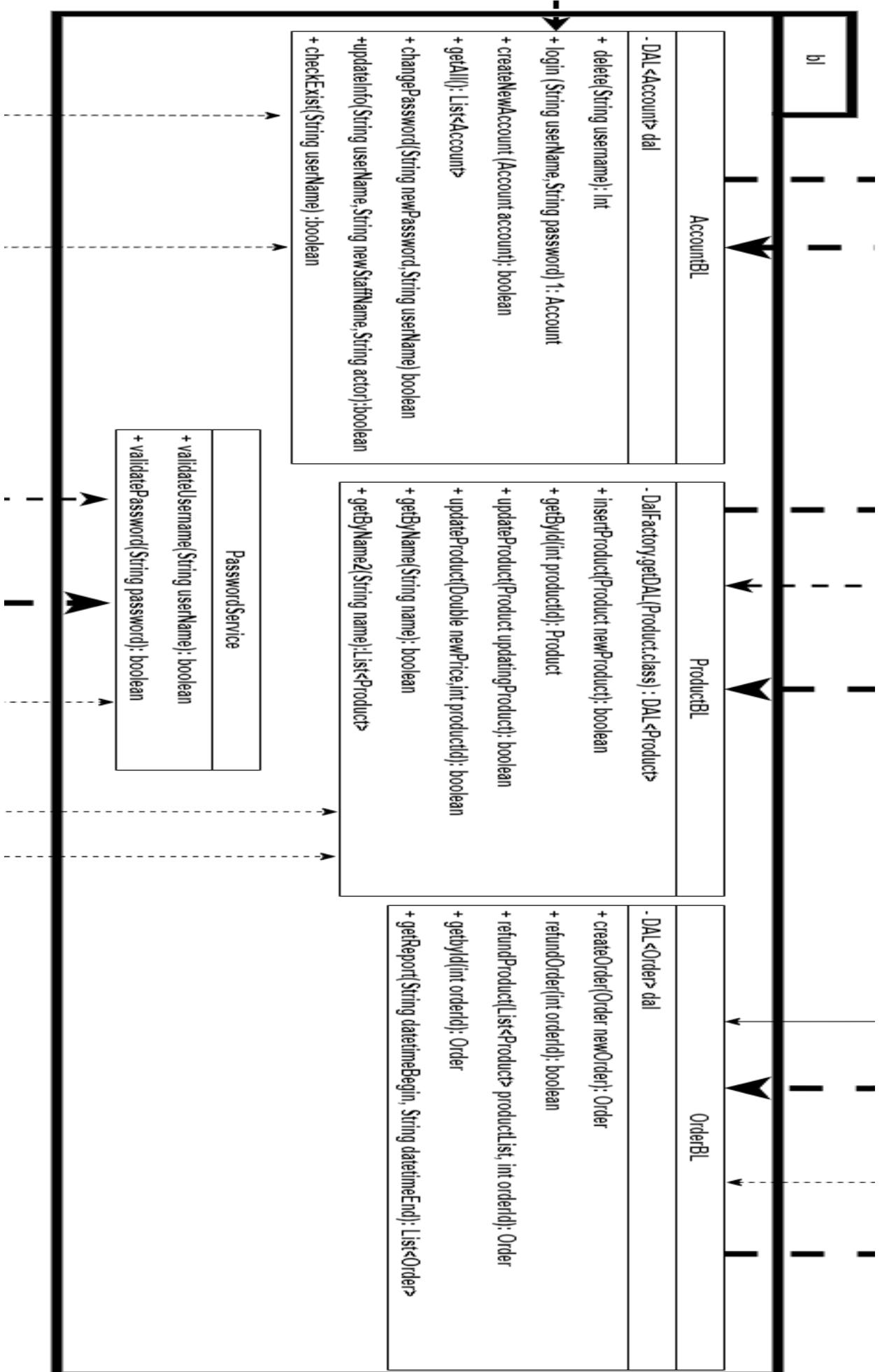
2. Code Design (Class Diagram)

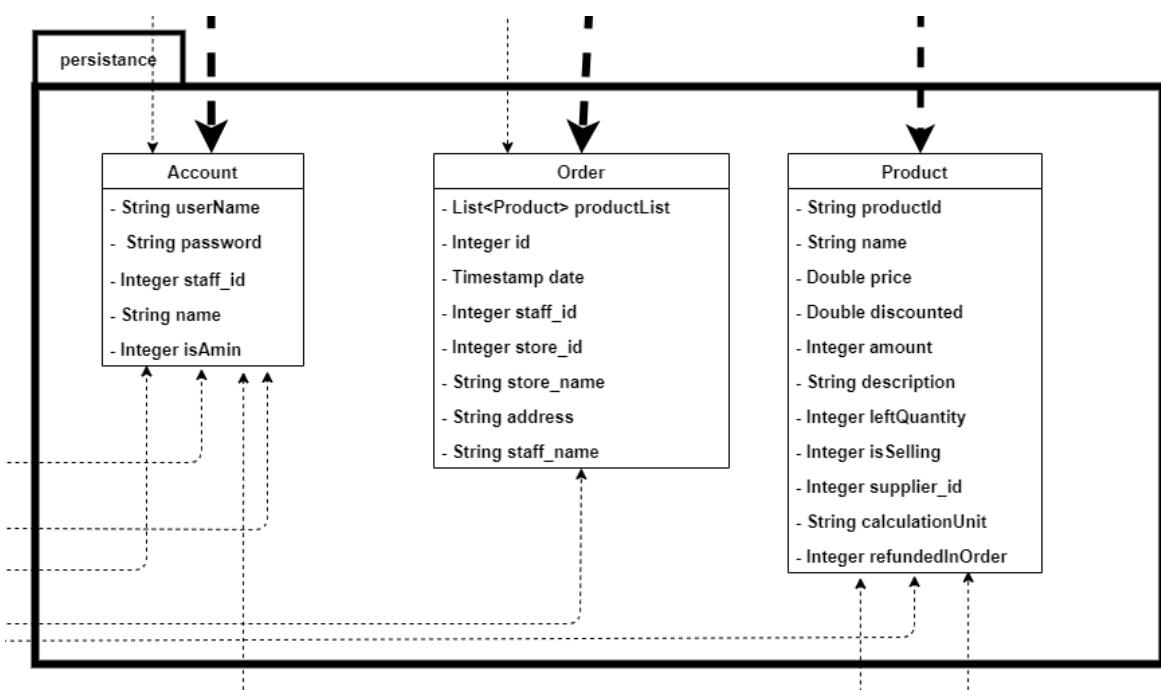
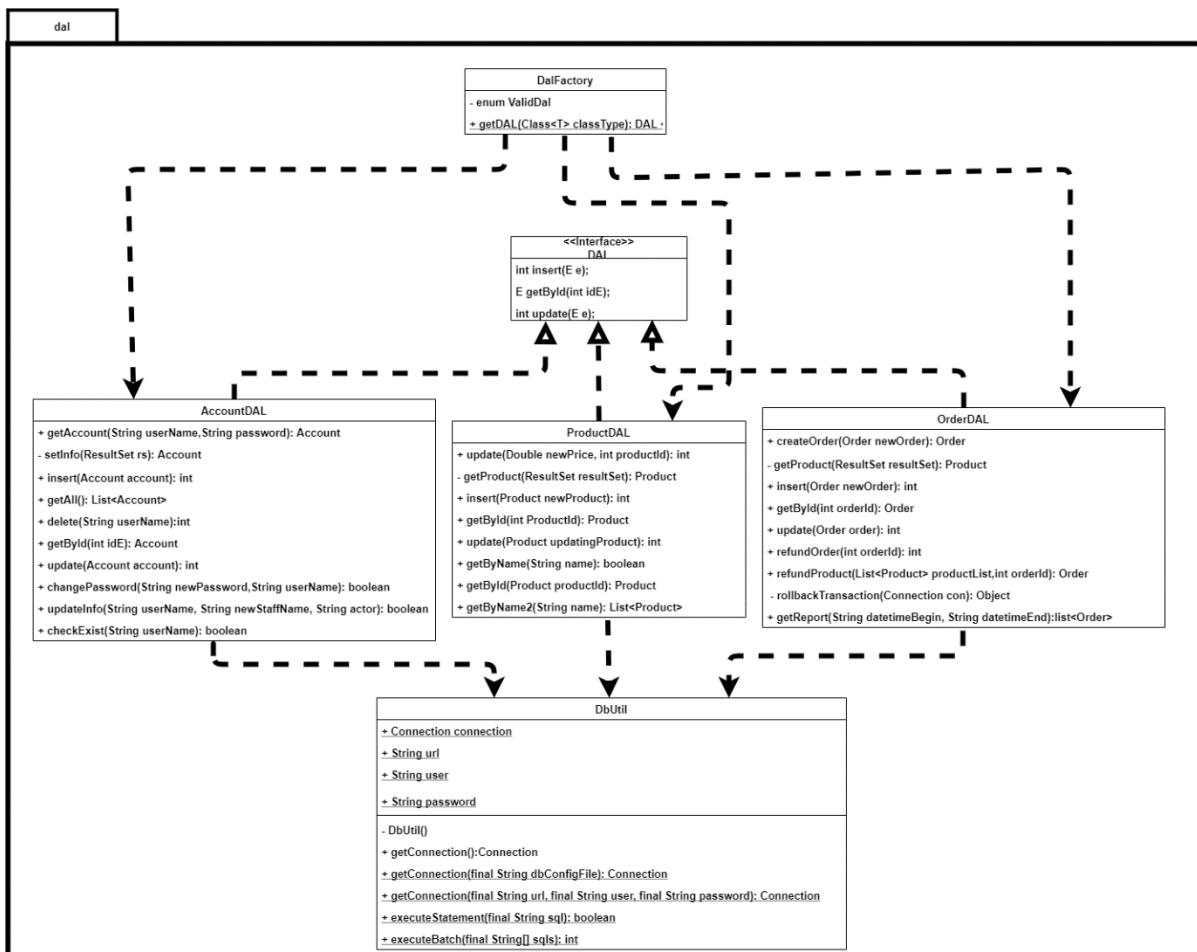
(Class Diagram):





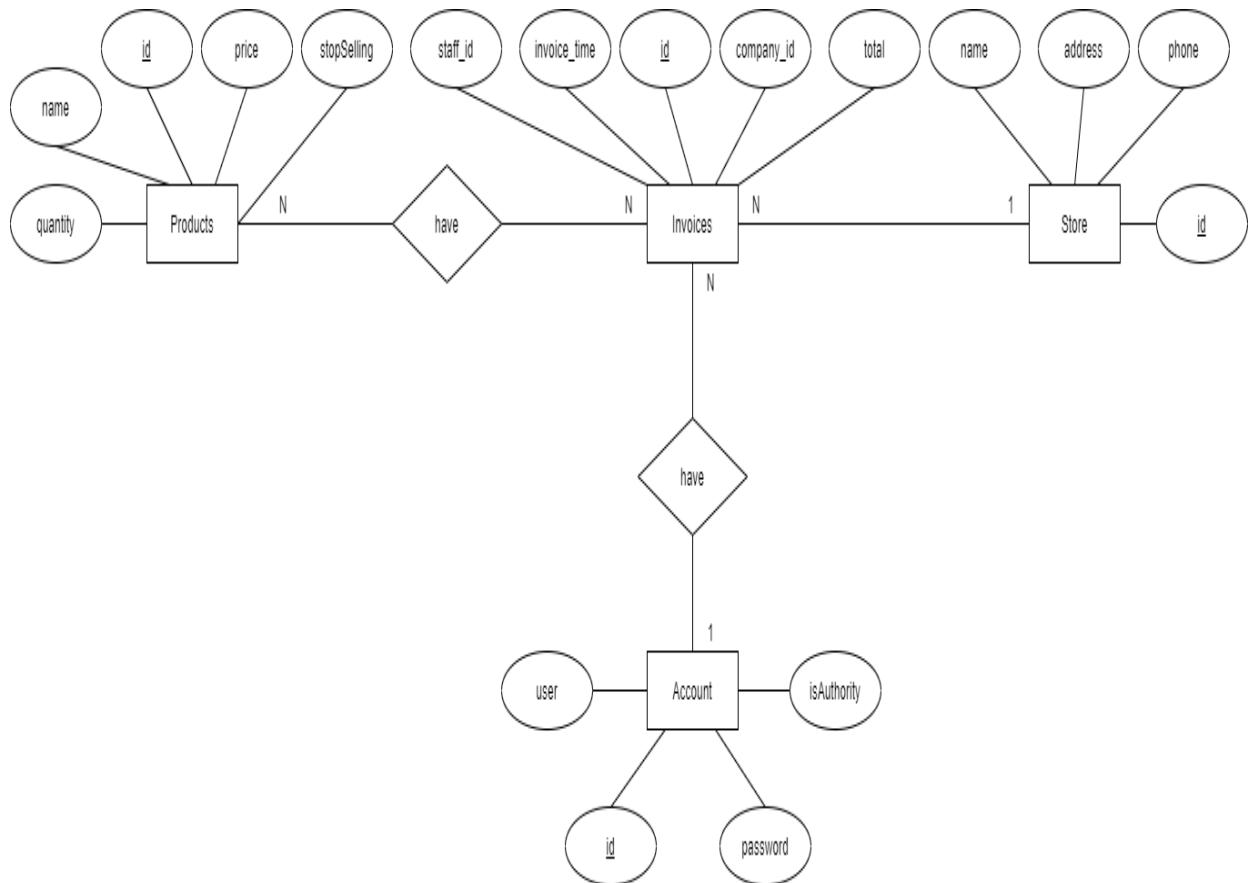




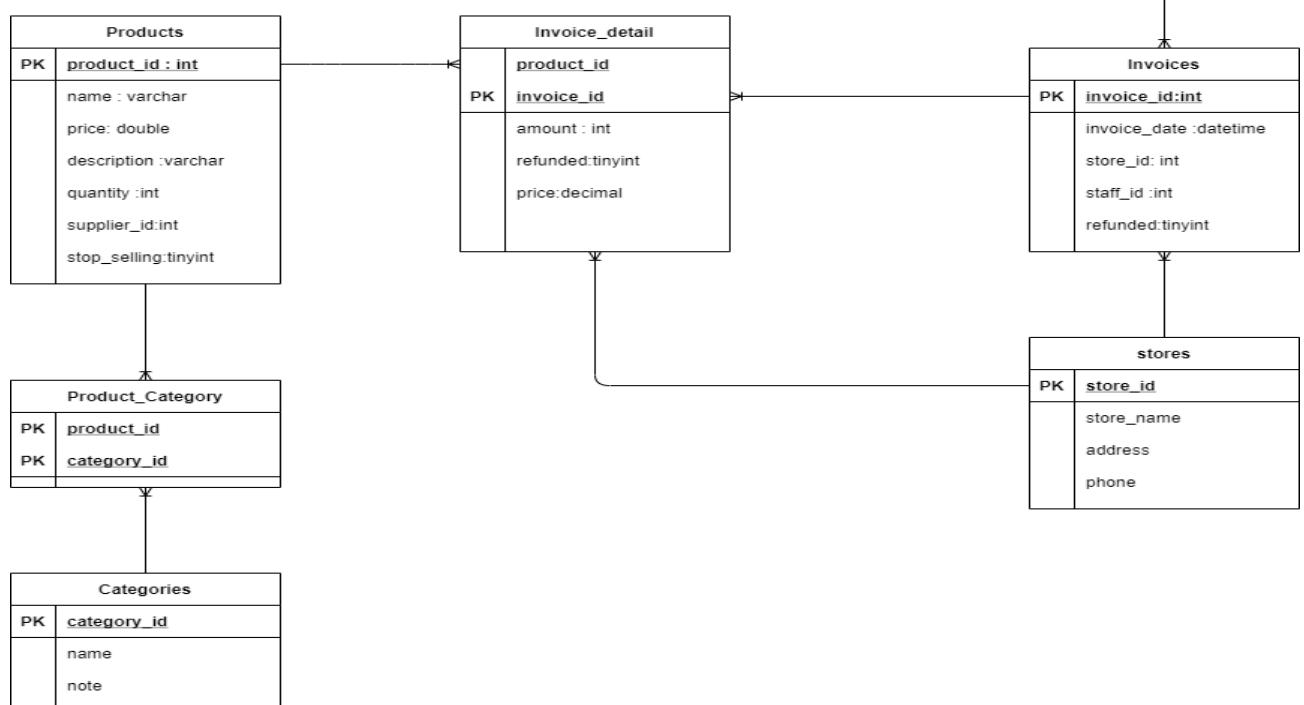


3. Database Design

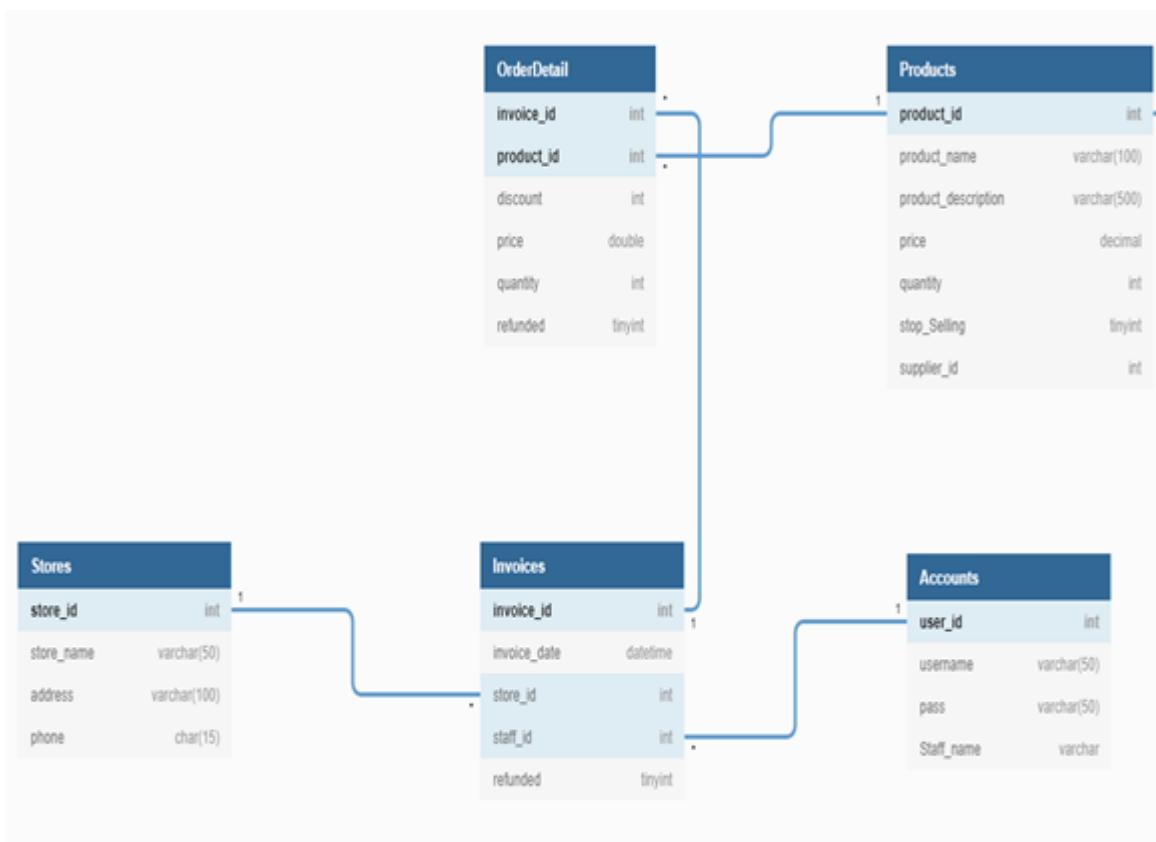
a. Entity Relationship Diagram



Account	
PK	staff_id
	user:varchar
	password:varchar
	name:varchar
	isAuthority:tinyint



b. Database Design Details



Products

Column name	Data Type	Constraints	Description
product_id	int	Primary key	
product_name	Varchar(100)	unique	
product_description	Varchar(500)		
price	decimal	>0	
quantity	int	>0	
stop_selling	tinyint	NOT NULL,default(1)	0: true 1: false
supplier_id	int	NOT NULL, FK	

Stores

Column name	Data Type	Constraints	Description
store_id	int	Primay key	
store_name	Varchar(100)	NOT NULL	
address	Varchar(200)		
phone	Char(15)		

Accounts

Column name	Data Type	Constraints	Description
user_id	int	Primay key ,NOT NULL	
user_name	varchar(50)	NOT NULL,unique	
pass	Varchar(50)	NOT NULL	

Invoices			
Column name	Data Type	Constraints	Description
Invoice_id	int	Primary key ,NOT NULL	
Invoice_date	datetime	NOT NULL	
Store_id	int	FK	
Staff_id	int	FK	
refunded	tinyint	NOT NULL, default(1)	0: true, 1: fasle

Order_Detail			
Column name	Data Type	Constraints	Description
Invoice_id	int	PK, FK, NOT NULL	
Product_id	int	PK, FK, NOT NULL	
price	double	>0	
quantity	int	>0	
refunded	tinyint	Default (1)	0: true, 1: false

IV. Test

1. Test login

Test CaseNumber	1.1 (getAccountTest1)
Test Case Name	Test login functionality
Test Case description	check the system's login ability
Preconditions	
Test Case Input	"Nguyenquyetthang", "16c0ce36e334e22fda8caca1b10c2f9c", "Nguyenquyetthang", null, 1, "thang", 0
Test Case expected output	True
Test Case Steps	Step1: create instance of accountDAL Step2: Login Step3: Create expected Account Step4: Check if the expected account is correct or not
Default Value preverving	

Test CaseNumber	1.2(getAccountTest2)
Test Case Name	Test login functionality
Test Case description	check the system's login ability
Preconditions	
Test Case Input	"staff", "123456"
Test Case expected output	True
Test Case Steps	Step1: create instance of accountDAL Step2: Login Step3: Create expected Account Step4: Check if the expected account is correct or not
Default Value preverving	

Test CaseNumber	1.3(getAccountTest3)
Test Case Name	Test login functionality
Test Case description	check the system's login ability
Preconditions	
Test Case Input	"staff1", "1234"
Test Case expected output	True
Test Case Steps	Step1: create instance of accountDAL Step2: Login Step3: Create expected Account Step4: Check if the expected account is correct or not
Default Value preverving	

Test CaseNumber	1.4(getAccountTest4)
Test Case Name	Test login functionality
Test Case description	check the system's login ability
Preconditions	
Test Case Input	"staff1", "1234"
Test Case expected output	True
Test Case Steps	Step1: create instance of accountDAL Step2: Login Step3: Create expected Account Step4: Check if the expected account is correct or not
Default Value preverving	

Test CaseNumber	1.5 (validdateUsername)
Test Case Name	Test username form
Test Case description	User Name mustn't have special character and It contains at least 8 characters and at most 20 characters
Preconditions	
Test Case Input	"Nguyenquyetthang"
Test Case expected output	True
Test Case Steps	Step1: Call validateUsername() Step2: Input value Step3: Test
Default Value preverving	

Test CaseNumber	1.6 (validdateUsername2)
Test Case Name	Test username form
Test Case description	Username have special character
Preconditions	
Test Case Input	“Nguyenquyetthang123%”
Test Case expected output	False
Test Case Steps	Step1: Call validateUsername() Step2: Input value Step3: Test
Default Value preverving	

Test CaseNumber	1.7(validdateUsername3)
Test Case Name	Test username from
Test Case description	Username less than 8 character
Preconditions	
Test Case Input	“Nguy”
Test Case expected output	False
Test Case Steps	Step1: Call validateUsername() Step2: Input value Step3: Test
Default Value preverving	

Test CaseNumber	1.8(validdateUsername4)
Test Case Name	Test username from
Test Case description	UserNmae more than 20 character
Preconditions	
Test Case Input	“Nguyenquyetthang123asdvgvghasdvgvghasvghasvdhv”
Test Case expected output	False
Test Case Steps	Step1: Call validateUsername() Step2: Input value Step3: Test
Default Value preverving	

Test CaseNumber	1.9(validatePassword)
Test Case Name	Test password
Test Case description	It contains at least 8 characters and at most 20 characters. It contains at least one digit. It contains at least one upper case alphabet. It contains at least one lower case alphabet. It doesn't contain any white space
Preconditions	
Test Case Input	"Thangnguenquyet123"
Test Case expected output	True
Test Case Steps	Step1: Call validatePassword() Step2: Input value Step3: Test
Default Value preverving	

Test CaseNumber	1.10 (validatePassword2)
Test Case Name	Test password
Test Case description	It wasn't contains at least one digit.
Preconditions	
Test Case Input	"Thangnguyenquyet"
Test Case expected output	False
Test Case Steps	Step1: Call validatePassword2() Step2: Input value Step3: Test
Default Value preverving	

Test CaseNumber	1.11(validatePassword3)
Test Case Name	Test password
Test Case description	It wasn't contains at least upper case alphabet
Preconditions	
Test Case Input	"thangnguyenquyet123"
Test Case expected output	False
Test Case Steps	Step1: Call validatePassword3() Step2: Input value Step3: Test
Default Value preverving	

Test CaseNumber	1.12(validdatePassword4)
Test Case Name	Test password functionality
Test Case description	It wans't contains at least lower case alphabet
Preconditions	
Test Case Input	"THANGNGUYENQUYET"
Test Case expected output	False
Test Case Steps	Step1: Call validatePassword4() Step2: Input value Step3: Test
Default Value preverving	

Test CaseNumber	1.13(validdatePassword5)
Test Case Name	Test password functionality
Test Case description	It contains at least 8 characters
Preconditions	
Test Case Input	"THA"
Test Case expected output	False
Test Case Steps	Step1: Call validatePassword4() Step2: Input value Step3: Test
Default Value preverving	

Test CaseNumber	1.14(validdatePassword6)
Test Case Name	Test password functionality
Test Case description	It contains at more 20 characters
Preconditions	
Test Case Input	"THAasdfasvgasbvghdvasghdasvasdvsavghasdgvdgasv"
Test Case expected output	False
Test Case Steps	Step1: Call validatePassword4() Step2: Input value Step3: Test
Default Value preverving	

Test CaseNumber	1.15(testMD5)
Test Case Name	Encrypt password
Test Case description	Ensure security for password
Preconditions	
Test Case Input	"Thangnguyenquyet123"
Test Case expected output	16c0ce36e334e22fda8caca1b10c2f9c
Test Case Steps	Step1: Call getMD5() Step2: Input value Step3: Create expected output Step4: Compare result
Default Value preverving	

2. Test order

Test CaseNumber	2.1(getByIdTest)
Test Case Name	Test order functionality
Test Case description	wrong orderId
Preconditions	
Test Case Input	1001
Test Case expected output	Null
Test Case Steps	Step1: Call getByIdTest() Step2: Input value Step3: Test
Default Value preverving	

Test CaseNumber	2.2(getByIdTest2)
Test Case Name	Test order functionality
Test Case description	orrect id
Preconditions	
Test Case Input	1
Test Case expected output	Null
Test Case Steps	Step1: Call getByIdTest2() Step2: Input value Step3: Test
Default Value preverving	

Test CaseNumber	2.3(createOrder)
Test Case Name	Test order functionality
Test Case description	correct
Preconditions	
Test Case Input	1001,"phobo",50000.,10000.,1
Test Case expected output	Null
Test Case Steps	Step1: Call createOrder() Step2: Input value Step3: Test
Default Value preverving	

Test CaseNumber	2.4(createOrder1)
Test Case Name	Test order functionality
Test Case description	order null
Preconditions	
Test Case Input	
Test Case expected output	Null
Test Case Steps	Step1: Call createOrder1() Step2: Input value Step3: Test
Default Value preverving	

Test CaseNumber	2.5(createOrder2)
Test Case Name	Test order functionality
Test Case description	wrong product id
Preconditions	
Test Case Input	10010,"phobo",50000.,10000.,1
Test Case expected output	Null
Test Case Steps	Step1: Call createOrder2() Step2: Input value Step3: Test
Default Value preverving	

Test CaseNumber	2.6(createOrder3)
Test Case Name	Test order functionality
Test Case description	List null
Preconditions	
Test Case Input	"Ha Noi"
Test Case expected output	Null
Test Case Steps	Step1: Call createOrder3() Step2: Input value Step3: Test
Default Value preverving	

Test CaseNumber	2.7(createOrder3)
Test Case Name	Test order functionality
Test Case description	List null
Preconditions	
Test Case Input	10010,"phobo",50000.,10000.,1
Test Case expected output	Null
Test Case Steps	Step1: Call createOrder3() Step2: Input value Step3: Test
Default Value preverving	

Test CaseNumber	2.8(refundProductTest)
Test Case Name	Test order functionality
Test Case description	List null
Preconditions	
Test Case Input	1001
Test Case expected output	Null
Test Case Steps	Step1: Call refundProductTest () Step2: Input value Step3: Test
Default Value preverving	

3. Test product

Test CaseNumber	3.1(getProductById)
Test Case Name	Test product functionality
Test Case description	correct id
Preconditions	
Test Case Input	1001,"phobo",20000.0,null,null,"da update",20,1,1,null
Test Case expected output	True
Test Case Steps	Step1: Call getProductById() Step2: Input value Step3: Test
Default Value preverving	

Test CaseNumber	3.2(getProductById2)
Test Case Name	Test product functionality
Test Case description	wrong id
Preconditions	
Test Case Input	1
Test Case expected output	Null
Test Case Steps	Step1: Call getProductById2() Step2: Input value Step3: Test
Default Value preverving	

Test CaseNumber	3.3(insertProduct)
Test Case Name	Test product functionality
Test Case description	Already existed
Preconditions	
Test Case Input	1001,"my goi 5",10000.,"update from java app",10,1,"my goi"
Test Case expected output	Fale
Test Case Steps	Step1: Call insertProduct() Step2: Input value Step3: Test
Default Value preverving	

Test CaseNumber	3.4(updatePrice)
Test Case Name	Test product functionality
Test Case description	valid price
Preconditions	
Test Case Input	50000.,1001
Test Case expected output	True
Test Case Steps	Step1: UpdatePrice() Step2: Input value Step3: Test
Default Value preverving	

Test CaseNumber	3.5(updatePrice2)
Test Case Name	Test product functionality
Test Case description	wrong price (price<0)
Preconditions	
Test Case Input	-50000.,1001
Test Case expected output	False
Test Case Steps	Step1: UpdatePrice2() Step2: Input value Step3: Test
Default Value preverving	

Test CaseNumber	3.6(updatePoduct)
Test Case Name	Test product functionality
Test Case description	update in desciption
Preconditions	
Test Case Input	1002,"banh my",15000.,"update from java app dbsjajdasjad",50,1,"my goi"
Test Case expected output	True
Test Case Steps	Step1: UpdateProduct() Step2: Input value Step3: Test
Default Value preverving	

Test CaseNumber	3.7(updatePoduct1)
Test Case Name	Test product functionality
Test Case description	update false: wrong id
Preconditions	
Test Case Input	1111,"banh my",15000.,"update from java app dbsjajdasjad",50,1,"my goi"
Test Case expected output	False
Test Case Steps	Step1: UpdateProduct1() Step2: Input value Step3: Test
Default Value preverving	

Test CaseNumber	3.8(updatePoduct2)
Test Case Name	Test product functionality
Test Case description	update false: wrong price (<0)
Preconditions	
Test Case Input	1002,"banh my",-15000.,"update from java app dbsjajdasjad",50,1,"my goi"
Test Case expected output	False
Test Case Steps	Step1: UpdateProduct2() Step2: Input value Step3: Test
Default Value preverving	

Test CaseNumber	3.9(updatePoduct3)
Test Case Name	Test product functionality
Test Case description	update false: duplicate name
Preconditions	
Test Case Input	1002,"phobo",15000.,"update from java app dbsjajdasjad",50,1,"my goi"
Test Case expected output	False
Test Case Steps	Step1: UpdateProduct3() Step2: Input value Step3: Test
Default Value preverving	

Test CaseNumber	3.10(updatePoduct4)
Test Case Name	Test product functionality
Test Case description	update false: null
Preconditions	
Test Case Input	productBL.updateProduct(product)
Test Case expected output	False
Test Case Steps	Step1: UpdateProduct4() Step2: Input value Step3: Test
Default Value preverving	

Test CaseNumber	3.11(getByName)
Test Case Name	Test product functionality
Test Case description	exist :True
Preconditions	
Test Case Input	"phobo"
Test Case expected output	True
Test Case Steps	Step1: getByName() Step2: Input value Step3: Test
Default Value preverving	

Test CaseNumber	3.12(getByName1)
Test Case Name	Test product functionality
Test Case description	exist :False
Preconditions	
Test Case Input	"phobo2"
Test Case expected output	True
Test Case Steps	Step1: getByName2() Step2: Input value Step3: Test
Default Value preverving	

V. Assign work to each team member

Group 4	Project Name					
No	Task name	Description	Start Date	End Date	Member	Self assessment
1	System design	System model architecture	24/07/2020		Nguyễn Quyết Thắng	
2	Use case design	Design system features	30/07/2020		Nguyễn Quyết Thắng, Trần Văn Huân, Nguyễn Vũ Huân	
3	Entity relationships design	Identify the entities in the system and the relationships between them	28/07/2020	29/07/2020	Trần Văn Huân Nguyễn Quyết Thắng	
4	Entity relationships design details	Construct tables that represent properties of entities and the relationship between entities	29/07/2020	04/08/2020	Nguyễn Quyết Thắng	

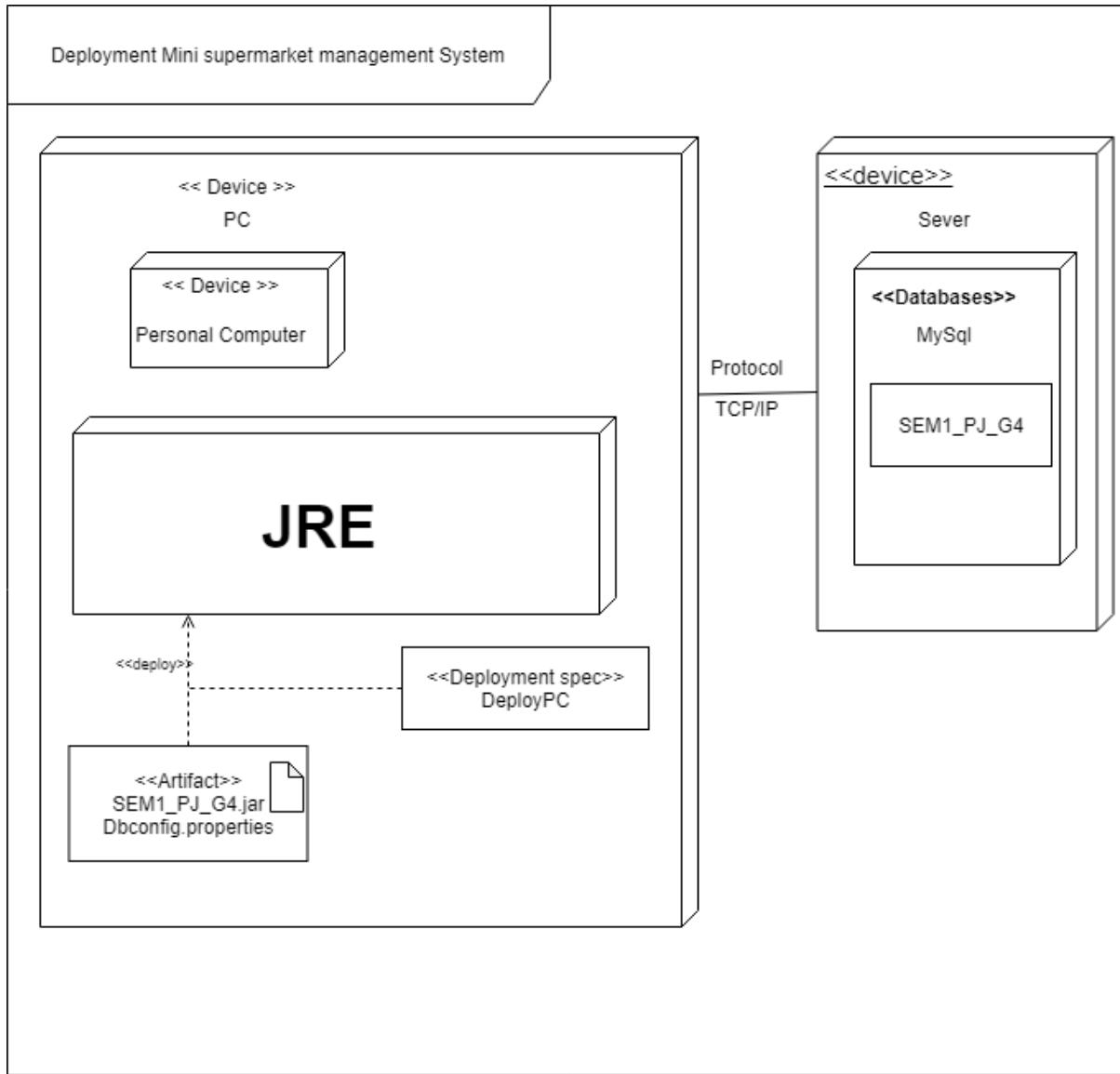
5	Design database	Database building	30/07/2020	06/08/2020	Nguyễn Quyết Thắng	
6	Use case description	Describe in detail the use case	01/08/2020	06/08/2020	Nguyễn Quyết Thắng Trần Văn Huân Nguyễn Vũ Huân	
7	Design activity diagram for each use case	Design the system's task processing flowchart for each use case	01/08/2020	07/08/2020	Nguyễn Vũ Huân Trần Văn Huân Nguyễn Quyết Thắng	
8	Design sequence diagram for each use case	Design sequence diagrams for the system's workflow for each use case	01/08/2020	25/08/2020	Trần Văn Huân Nguyễn Vũ Huân Nguyễn Quyết Thắng	
9	Class diagram for system	Designing detailed drawing class for the system	03/08/2020	20/08/2020	Trần Văn Huân Nguyễn Vũ Huân Nguyễn Quyết Thắng	
10	Coding	Write code based on class diagrams	15/08/2020	26/8/2020	Nguyễn Quyết Thắng	

					Trần Văn Huân Nguyễn Vũ Huân	
11	Bug fixes, optimizations, upgrades for the code	Fix errors, optimize code, upgrade system features	15/8/2020		Nguyễn Quyết Thắng	
12	Deployment diagram	Drawing up the project environment	01/08/2020	26/08/2020	Nguyễn Quyết Thắng Trần Văn Huân Nguyễn Vũ Huân	
13	Attach report	Report writing	07/08/2020	26/08/2020	Nguyễn Vũ Huân Trần Văn Huân Nguyễn Quyết Thắng	
14	Wriet slides	Presentation material	26/08/2020	26/08/2020	Nguyễn Vũ Huân Trần Văn Huân Nguyễn Quyết Thắng	

VI. Installation Instructions

1. Deployment Diagram

Deployment diagram of a mini market management System



Introduction for using the app:

- Install JDK:
 - Follow link:
<https://www.oracle.com/technetwork/java/javase/downloads/index.html>
Check: Open command line => Type commands: java --version
If check don't version:
 - + Copy: "C:\Program Files\Java\jdk-13.0.1\bin" add to "Path" in Enviroment Variables
 - + Double-click to "This Computer, chọn Properties -> Advanced System Settings -> Enviroment Variables"
- Install Maven:
 - Setting the environment variable JAVA_HOME.
 - Install Apache maven

- Setting the environment variable Maven
- Install Maven:
 - + Follow link: apache-maven-3.6.3-bin.zip
 - + Unzip new create file:

apache-maven-3.6.3

```
| -- bin
  | -- m2.conf
  | -- mvn
  | -- mvn.cmd
  | -- mvnDebug
  | -- mvnDebug.cmd
  | -- mvnyjp
| -- boot
| -- conf
| -- lib
| -- LICENSE
| -- NOTICE
| -- README.txt
```

- + Setting “Path” to folder /apache-maven-3.6.3/bin
- + Check setting success:
 - Run app Command Prompt/PowerShell in Windows
 - Run app Terminal in Unix/Linux/macOS
 - Type commands: “mvn –version”
- Install MySQL:
 1. Installing MySQL on Windows
 - + Download MySQL ZIP ARCHIVE from
<https://dev.mysql.com/downloads/mysql/>
 - Choose "General Available (GA) Releases" tab.
 - Under "MySQL Community Server 5.7.xx" => In "Select Platform", choose "Microsoft Windows".
 - Under "Other Downloads", download "Windows (x86, 64-bit), ZIP ARCHIVE (mysql-5.7.xx-winx64.zip)" or "Windows (x86, 32-bit), ZIP ARCHIVE (mysql-5.7.xx-win32.zip)".
 - (You can check whether your Windows is 32-bit or 64-bit from "Control Panel" => System and Security (optional) => System => Under "System Type". I doubt that there are still 32-bit Windows around!)
 - There is NO need to "Login" or "Sign up" - Just click "No thanks, just start my downloads!"

+ UNZIP the downloaded file into your project directory "C:\myWebProject". MySQL will be unzipped as "c:\myWebProject\mysql-5.7.{xx}-winx64". For EASE OF USE, we shall shorten and rename the directory to "c:\myWebProject\mysql". Take note and remember your MySQL installed directory!!!

+ (NEW since MySQL 5.7.7) Initialize the database: Start a CMD (as administrator) ("Search" button => Enter "cmd" => Right-Click => Run as Administrator) and issue these commands:

```
// Change directory to the MySQL's binary directory
// Suppose that your MySQL is installed in "c:\myWebProject\mysql"
c:
cd \myWebProject\mysql\bin
// Initialize the database. Create a root user with random password. Show the messages on console
mysql --initialize --console
.....
..... [Note] A temporary password is generated for root@localhost: xxxxxxxx
```

During the installation, a superuser called root is created with a temporary password, as shown above. TAKE NOTE of the PASSWORD, COPY and save it somewhere, and TAKE A PICTURE!!!!

+ If you make a mistake somewhere or forgot your password, DELETE the entire MySQL directory "C:\myWebProject\mysql", and REPEAT step 2 and 3

2. Installing MySQL on macOS

+ Download the MySQL "DMG Archive" from
<https://dev.mysql.com/downloads/mysql/>:

- Choose "General Available (GA) Releases" tab.
- Under "MySQL Community Server 5.7.xx" => In "Select Platform", choose the "Mac OS X".
- Select the appropriate "DMG Archive" for your specific Mac OS version, IF more than one versions are available. Otherwise, you have no other choices!

To check your OS version => Click the 'Apple' logo => "About this Mac".

To check whether your Mac OS is 32-bit or 64-bit => Read <http://support.apple.com/kb/ht3696>. Unless you have a dinosaur-era machine, it should be 64-bit!

- There is NO need to "Login" or "Sign up" - Just click "No thanks, just start my download"

+ To install MySQL:

- Go to "Downloads" => Double-click ".dmg" file downloaded.
- Double-click the "mysql-5.7.{xx}-osx10.x-xxx.pkg"
- Follow the screen instructions to install MySQL. During the installation, a superuser called root is created with a temporary random password. TAKE NOTE of the PASSWORD, COPY and save

it somewhere, and TAKE A PICTURE!!!. For the latest MySQL, password is sent to the notifications as well.

- MySQL will be installed in "/usr/local/mysql". Take note of this installed directory!!
- Eject the ".dmg" file

+ If you make a mistake somewhere or forgot your password, stop the server (Click "Apple" Icon => System Preferences => MySQL => Stop).

Goto /usr/local (via Finder => Go => GoTo Folder => type /usr/local) and remove all the folders beginning with "mysql...", e.g., "mysql-5.7.{xx}..." and "mysql", and Re-run Step 2.

+ Start/Stop MySQL

```
sudo /usr/local/mysql/support-files/mysql.server start
```

```
sudo /usr/local/mysql/support-files/mysql.server stop
```

3. Installing MySQL on Linux/UNIX

+ Install MySQL (Open a new Terminal and issue this command):

```
$ sudo apt-get update
```

```
$ sudo apt-get install mysql-server
```

+ Verify the MySQL Installation

```
// Show packages relating to mysql
$ dpkg --get-selections | grep mysql

// Check the details of a package
$ dpkg --status mysql-server

// List the installed files of a package
$ dpkg --listfiles mysql-server

// Check the location of "mysqld" (MySQL server deamon)
$ which mysqld
$ whereis mysqld
$ man mysqld // Read the manual

// Check the location of "mysql" (MySQL command-line client)
$ which mysql
$ whereis mysql
$ man mysql // Read the manual
```

+ Configure MySQL Server

MySQL reads the startup options from the files shown below, in the specified order (top files are read first, files read later take precedence)

(Reference: <http://dev.mysql.com/doc/refman/5.7/en/option-files.html>)

```

/etc/my.cnf
/etc/mysql/my.cnf
SYSCONFDIR/my.cnf
$MYSQL_HOME/my.cnf (server only)
//The file specified in --defaults-extra-file startup option, if any
~/.my.cnf
~/.mylogin.cnf (client only)

```

The installation default /etc/mysql/my.cnf includes directories /etc/mysql/conf.d/ and /etc/mysql/mysql.conf.d/. The /etc/mysql/conf.d/mysql.cnf is empty. Hence, the main configuration file is /etc/mysql/mysql.conf.d/mysqld.cnf. Browse through /etc/mysql/mysql.conf.d/mysqld.cnf:

+ Configure MySQL Server

Browse through /etc/mysql/mysql.conf.d/mysqld.cnf:

```

[mysqld]
user          = mysql
port          = 3306
basedir       = /usr
datadir       = /var/lib/mysql
.....
log_error     = /var/log/mysql/error.log
.....

```

A special user called "mysql" is created to run the MySQL server.

The server runs on the default port number of 3306.

The data directory is located at /var/lib/mysql (owned by mysql:mysql).

The error log is located at /var/log/mysql/error.log.

+ Start/Shutdown MySQL Server (mysqld)

MySQL is run as a service called "mysql" (configured at "/etc/init.d/mysql"), which is started automatically after boot. To start/stop/restartmysql, you could:

```

sudo service mysql start
sudo service mysql stop
sudo service mysql restart
sudo service mysql status

```

MySQL server will be started automatically after the installation. You CANNOT start another instance! To check the mysqld process:

```
ps aux | grep mysqld
```

+ Start/Stop MySQL Command-line Client (mysql)

To start a MySQL client:

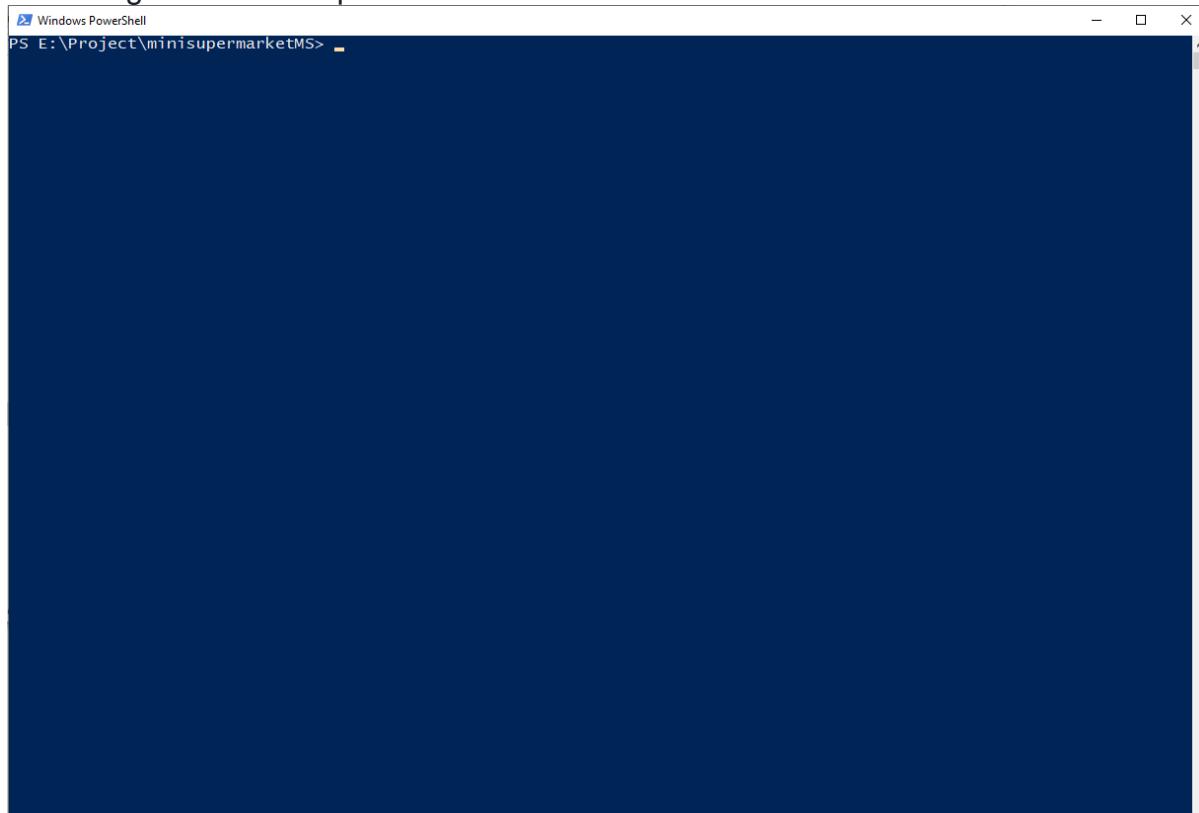
```
$ mysql -u root -p
// Enter the password for the root user you have set in Step 1
mysql> select user, host, authentication_string from mysql.user;
+-----+-----+-----+
| user | host | authentication_string |
+-----+-----+-----+
| root | localhost | *1556A6F65259CE3FBEA8489096F7797B4E0D5BEC |
| mysql.sys | localhost | *THISISNOTAVALIDPASSWORDTHATCANBEUSEDHERE |
| debian-sys-maint | localhost | *4DAD7B857498382CA8EB2EEF56E0EF9269D9FF6B |
+-----+-----+-----+
3 rows in set (0.00 sec)
mysql> exit
```

- Install App:
 - + Follow link: github.com/ngonngay/Project
 - + Download file ZIP
 - + Unzip file
 - + Biên dịch file:

Open folder “minisupermarketMS”

Name	Date modified	Type	Size
minisupermarketMS	9/15/2020 5:30 PM	File folder	
server	9/14/2020 3:43 PM	File folder	
testjavaapp	9/14/2020 3:43 PM	File folder	
.gitignore	9/14/2020 3:43 PM	Text Document	1 KB
LICENSE	9/14/2020 3:43 PM	File	2 KB
README.md	9/14/2020 3:43 PM	MD File	1 KB

Shift + right mouse to open PowerShell



Type commands “mvn package”

```

PS E:\Learning\Semeter1\ProjectSemeter1\Project\minisupermarketMS> mvn package
[INFO] Scanning for projects...
[INFO] ------------------------------------------------------------------------
[INFO] Building minisupermarketMS 1.0-SNAPSHOT
[INFO]   -- [ jar ] --
[INFO] --- maven-resources-plugin:3.0.2:resources (default-resources) @ minisupermarketMS ---
[INFO]   Using 'UTF-8' encoding to copy filtered resources.
[INFO]   skip non existing resourceDirectory E:\Learning\Semeter1\ProjectSemeter1\Project\minisupermarketMS\src\main\resources
[INFO] --- maven-compiler-plugin:3.8.0:compile (default-compile) @ minisupermarketMS ---
[INFO]   Nothing to compile - all classes are up to date
[INFO] --- maven-resources-plugin:3.0.2:testResources (default-testResources) @ minisupermarketMS ---
[INFO]   Using 'UTF-8' encoding to copy filtered resources.
[INFO]   skip non existing resourceDirectory E:\Learning\Semeter1\ProjectSemeter1\Project\minisupermarketMS\src\test\resources
[INFO] --- maven-compiler-plugin:3.8.0:testCompile (default-testCompile) @ minisupermarketMS ---
[INFO]   Nothing to compile - all classes are up to date
[INFO] --- maven-surefire-plugin:2.22.1:test (default-test) @ minisupermarketMS ---
[INFO]
[INFO] --- T E S T S

```

Type commands “mvn clean dependency:copy-dependence package”

```

PS E:\Learning\Semeter1\ProjectSemeter1\Project\minisupermarketMS> mvn clean dependency:copy-dependence package
[INFO] Scanning for projects...
[INFO] ------------------------------------------------------------------------
[INFO] Building minisupermarketMS 1.0-SNAPSHOT
[INFO]   -- [ jar ] --
[INFO] --- maven-clean-plugin:3.1.0:clean (default-clean) @ minisupermarketMS ---
[INFO] Deleting E:\Learning\Semeter1\ProjectSemeter1\Project\minisupermarketMS\target
[INFO] --- maven-dependency-plugin:2.8:copy-dependencies (default-cli) @ minisupermarketMS ---
[INFO]   Copying junit-4.11.jar to E:\Learning\Semeter1\ProjectSemeter1\Project\minisupermarketMS\target\dependency\junit-4.11.jar
[INFO]   Copying scala-library-2.13.0.jar to E:\Learning\Semeter1\ProjectSemeter1\Project\minisupermarketMS\target\dependency\scala-library-2.13.0.jar
[INFO]   Copying mysql-connector-java-8.0.20.jar to E:\Learning\Semeter1\ProjectSemeter1\Project\minisupermarketMS\target\dependency\mysql-connector-java-8.0.20.jar
[INFO]   Copying hamcrest-core-1.3.jar to E:\Learning\Semeter1\ProjectSemeter1\Project\minisupermarketMS\target\dependency\hamcrest-core-1.3.jar
[INFO]   Copying protobuf-java-3.6.1.jar to E:\Learning\Semeter1\ProjectSemeter1\Project\minisupermarketMS\target\dependency\protobuf-java-3.6.1.jar
[INFO]
[INFO] --- maven-resources-plugin:3.0.2:resources (default-resources) @ minisupermarketMS ---
[INFO]   Using 'UTF-8' encoding to copy filtered resources.
[INFO]   skip non existing resourceDirectory E:\Learning\Semeter1\ProjectSemeter1\Project\minisupermarketMS\src\main\resources
[INFO]
[INFO] --- T E S T S

```

After open folder “target”

Name	Date modified	Type	Size
.idea	9/15/2020 5:19 PM	File folder	
.settings	9/14/2020 3:43 PM	File folder	
.vscode	9/15/2020 2:50 PM	File folder	
src	9/14/2020 3:43 PM	File folder	
target	9/15/2020 4:47 PM	File folder	
xxx	9/15/2020 12:50 PM	File folder	
.classpath	9/14/2020 3:43 PM	CLASSPATH File	2 KB
.project	9/14/2020 3:43 PM	PROJECT File	1 KB
DB	9/15/2020 5:30 PM	SQL Text File	4 KB
minisupermarketMS.iml	9/14/2020 3:43 PM	IML File	2 KB
orderDAL	9/14/2020 3:43 PM	SQL Text File	1 KB
pom	9/14/2020 3:43 PM	XML Document	4 KB
query	9/15/2020 2:50 PM	SQL Text File	4 KB

Double-click to app icon

<input type="checkbox"/> Name	Date modified	Type
classes	9/17/2020 12:13 AM	File folder
dependency	9/17/2020 12:13 AM	File folder
generated-sources	9/17/2020 12:13 AM	File folder
generated-test-sources	9/17/2020 12:13 AM	File folder
maven-archiver	9/17/2020 12:13 AM	File folder
maven-status	9/17/2020 12:13 AM	File folder
surefire-reports	9/17/2020 12:13 AM	File folder
test-classes	9/17/2020 12:13 AM	File folder
<input checked="" type="checkbox"/> minisupermarketMS-1.0-SNAPSHOT.jar	9/17/2020 12:13 AM	Executable Jar File

VII. Appendix

Terms and abbreviation

<None>

References

<https://hararetail.com/>

<https://docs.oracle.com/javase>

<https://sinhnx.dev/>

Trung tâm Java Master

Channel youtube:

<https://www.youtube.com/channel/UCxNUqpgRkWnFiiRD0KcDvlw>

Vinmart

....

Issues

--Results--

+Create a product that combines the knowledge of the first semester 1

+Practice teamwork skill

+Familiarize with object oriented analysis design

--Limitations--

+Professional scope of the project

+Algorithm

....

--Experiences--

+Teamwork skill

+Familiarize with object oriented analysis design