|  |
| --- |
| **PROJECT REPORT**  Mini Supermarkets  Semester : Programming Fundamentals  Class : PF08    Group : Group 4  Instructor : Nguyễn Xuân Sinh  Team Menbers : 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 3

III. Design Details 16

IV. Test 22

V. Assign work to each team member 23

VI. Installation Instructions 24

I. Project Introduction

Process payment for orders in mini supermarkets

1. Proposed System

Actor Mannager

Actor Casher

Mannager function

1. Insert product
2. Update product
3. Accounts Management
4. Orders Management

Cashier function

1. Create order
2. Update 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

My SQL WorkBench

IntelliJ IDEA

Window 10

JDK13

6. Customer Requirements

Customers request (Feature system)

1- Application includes 6 functions including login, invoice creation, product

editing,add products, update products, manage sales invoices, manage

account.

- Login function is used to access the system

- Order creation function helps staff to pay for customers with full

information stores, selected products

- Update Order function helps receiver update information for order,

helps receiver to correct information for previously created order.

- Add a functional product to be used to add a new product to the system

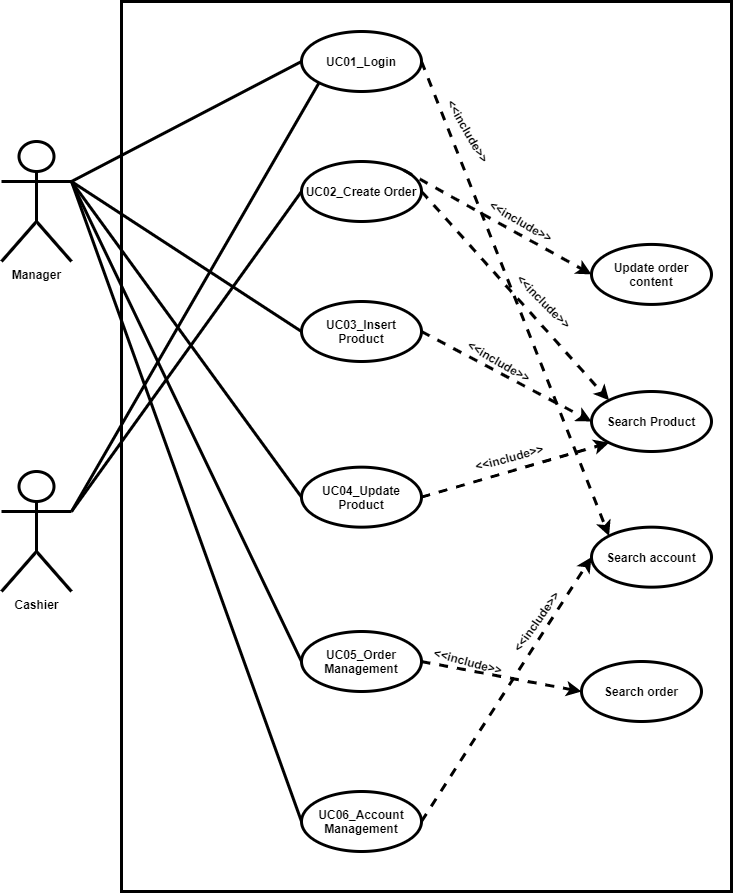
- Sales invoice management function used to statistic store sales

- Account manager function used to change the kernel name Change

passwordand name of kernel

II. Analyze System Requirements

1. Use case

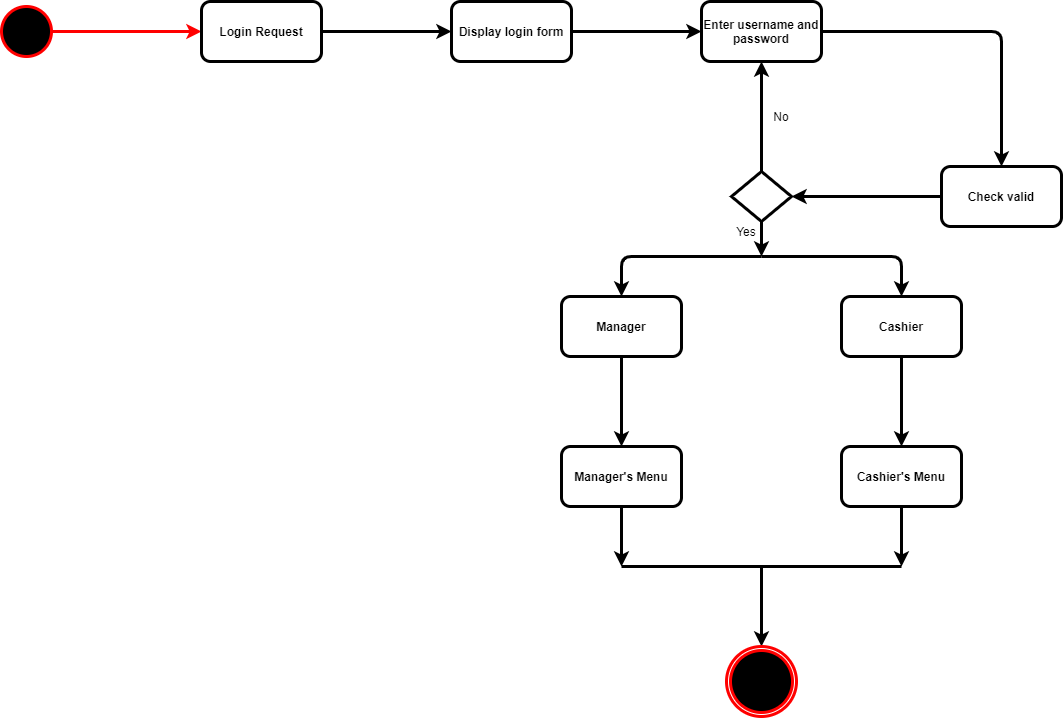


Use Case diagram

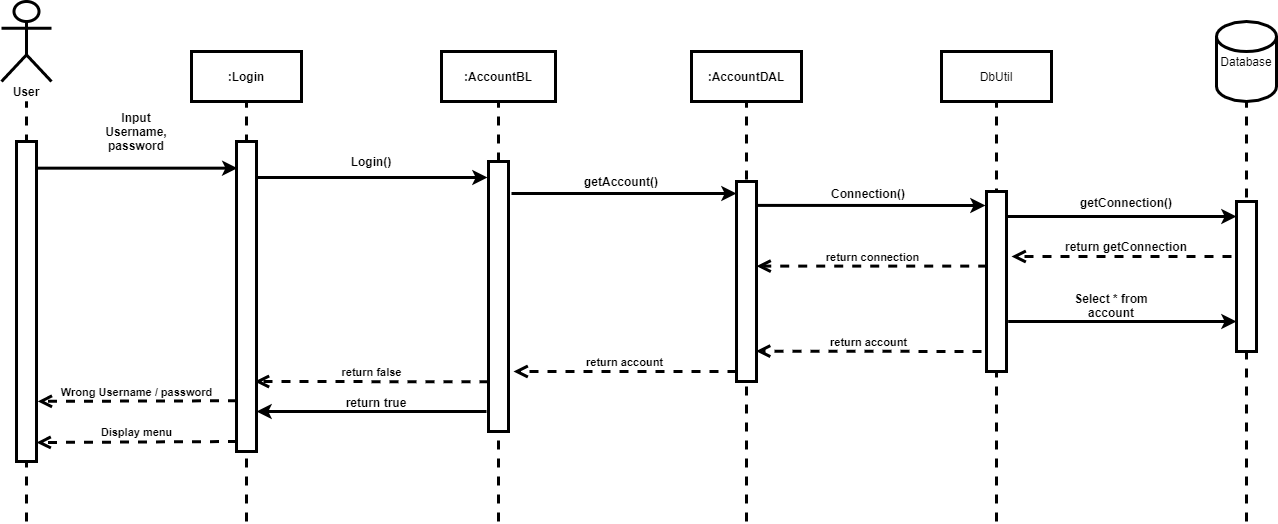
UC01\_Login (Description)

|  |  |
| --- | --- |
| 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.  2.1Valid 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)



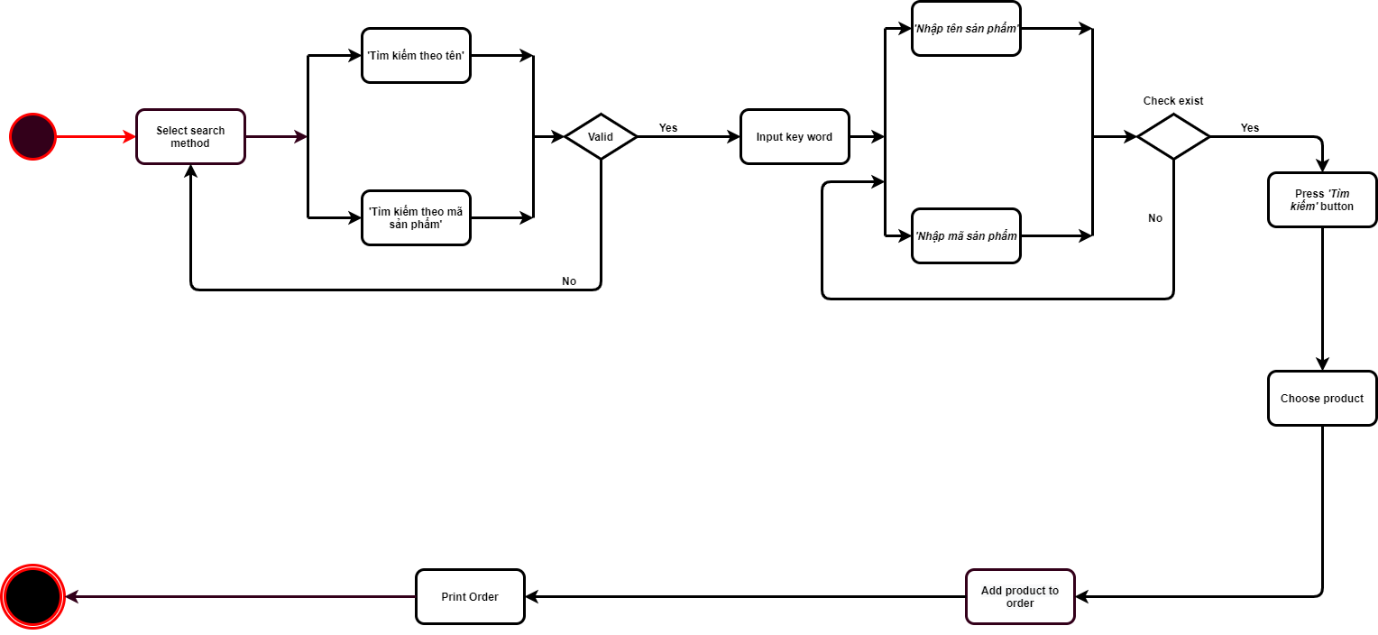
UC01\_Login (Sequence Diagram)



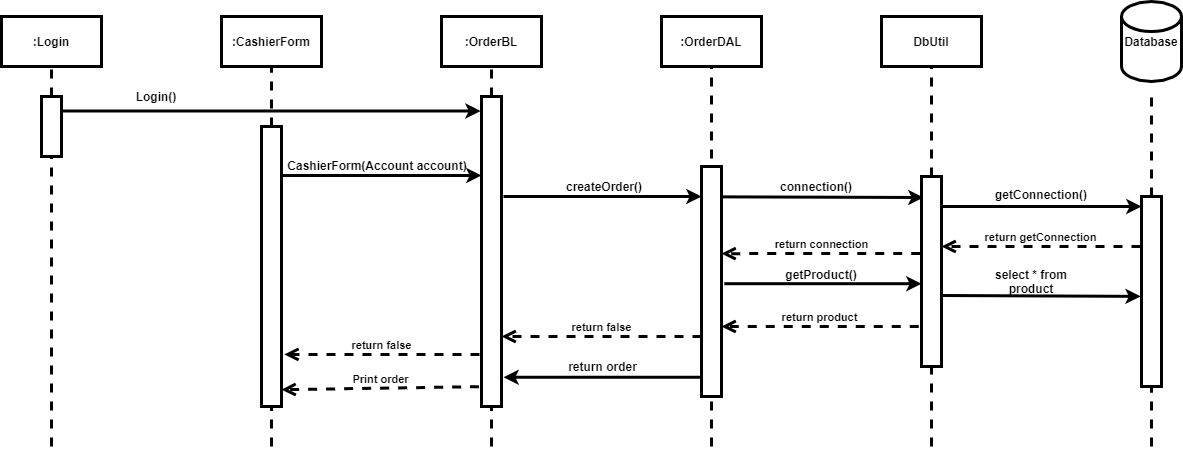
UC02\_Create Order (Description)

|  |  |
| --- | --- |
| 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 | 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 | AC1.   1. Search method    * 1. *‘Tìm kiếm theo tên’*      2. *‘Tìm kiếm theo mã sản phẩm’*   2. Valid search method  2.1.1 If user not search form yet  *-* Prompt user: *‘Hãy chọn phương thức tìm kiếm’*  AC2:  2.1 ‘*Nhập tên sản phẩm’*  2.2 ‘*Nhập mã sản phẩm’*  AC4:  *4.1*. Click left mouse to product  *4.2.* Click right mouse to product  AC5:  5*.*1: Press *‘Thêm vào hóa đơn’* button  5.2: Press *‘Yes’* button  5.3: Input *‘Số lượng’*  5.4: Press *‘ok’* button |
| Exception | EX 1: Product doesn’t exist   1. System prompt user: Product doesn’t exist in system   EX3: Quantity < = 0   1. Return user Main Course Step 4 |

UC02\_Create Order (Activity Diagram)



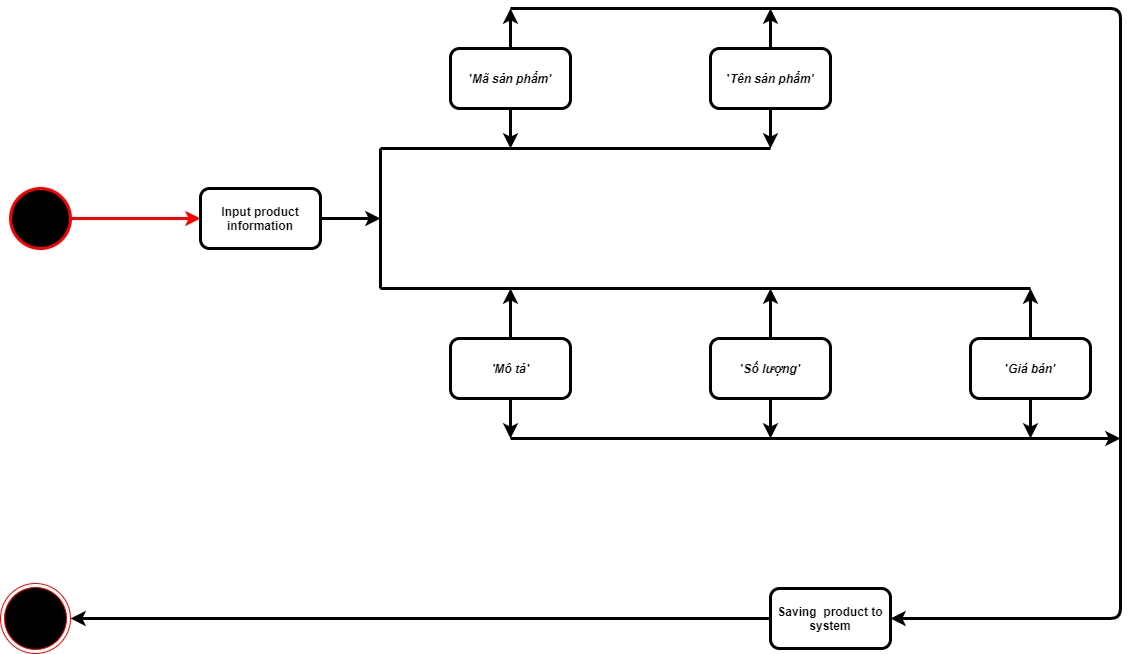
UC02\_Create Order (SequenceDiagram)



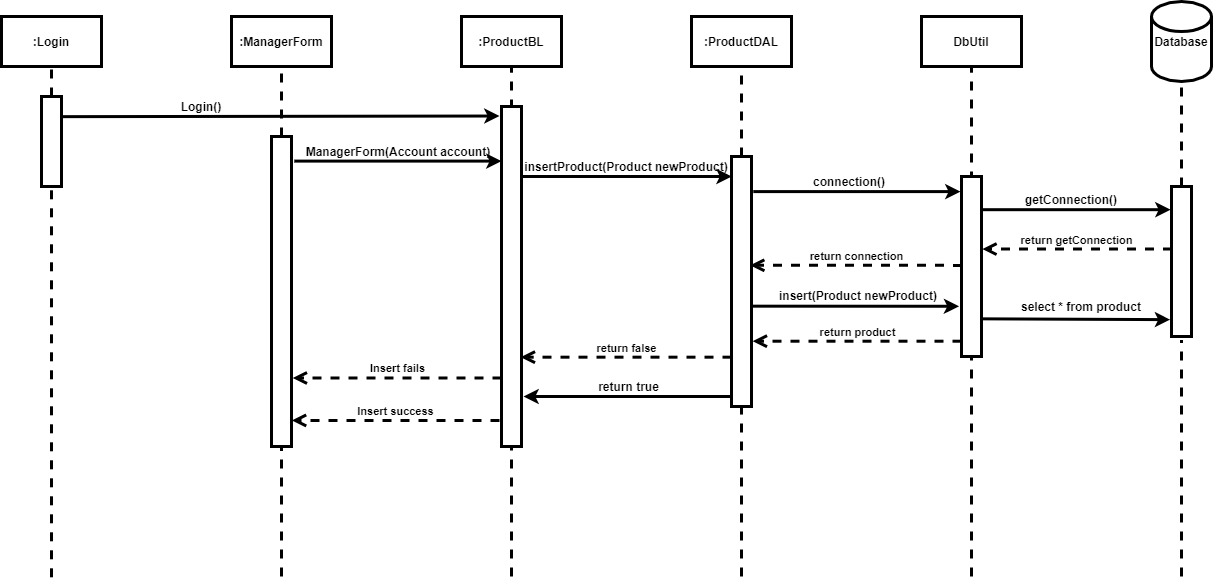
UC03\_Update Order (Description)

|  |  |
| --- | --- |
| 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 | 1. Input product information (AC1) 2. Add product to system 3. Click *‘ok’* 4. Saving to system |
| Alternate Courses | AC1: Input product information:   1. Input *‘Mã sản phẩm’*  * Click *‘Kiểm tra’* to check exist  1. Input *‘Tên sản phẩm’*  * Click *‘Kiểm tra’* to check exist  1. Input *‘Mô tả’* 2. Input *‘Số lượng’* 3. Input *‘Đơn vị’* 4. Input *‘Giá bán’* |
| Exception | EX1: Product does existed   1. Return user to Main courses step 1   EX2.1: Valid Name product   1. Name does existed  * Prompt user: *‘Sản phẩm này đã tồn tại trong hệ thống’*   - Return Main course 2  EX2.2: Valid *‘Số lượng’*  3.1. Quantity > 0  EX2.3: Valid *‘Giá bán’*  2.1. Price > 0 |

UC03\_Update Order (Activity Diagram)



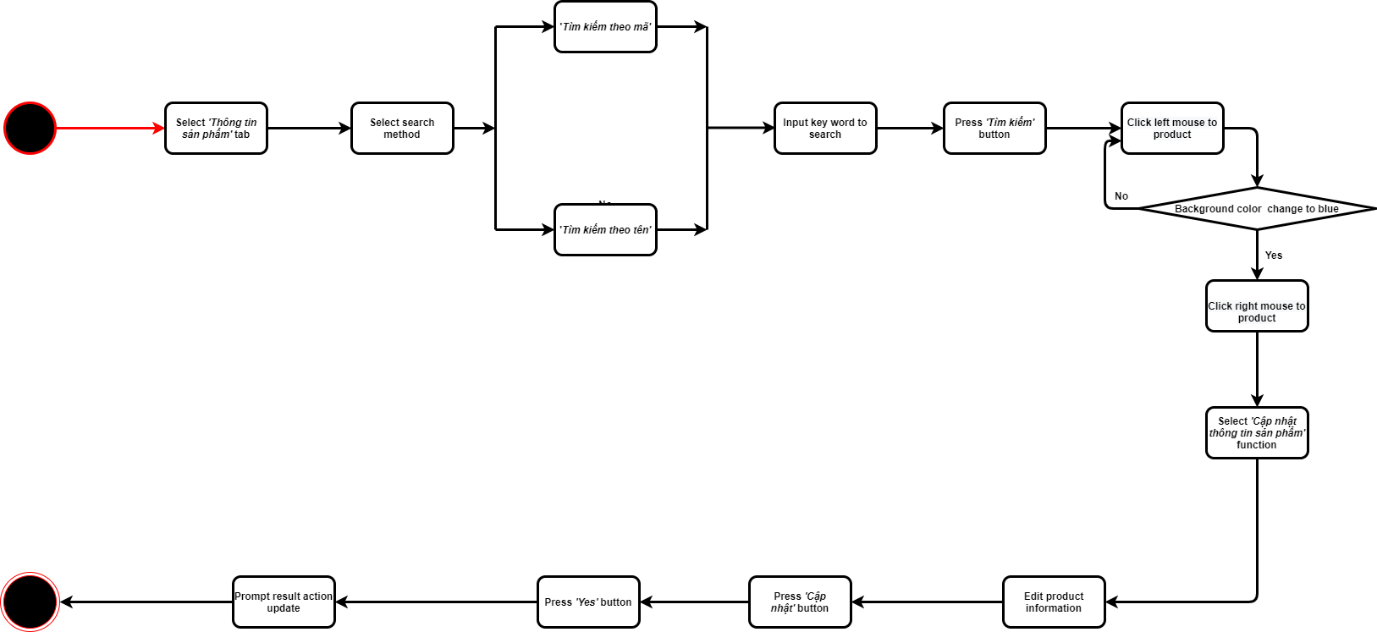
UC03\_Update Order (SequenceDiagram)



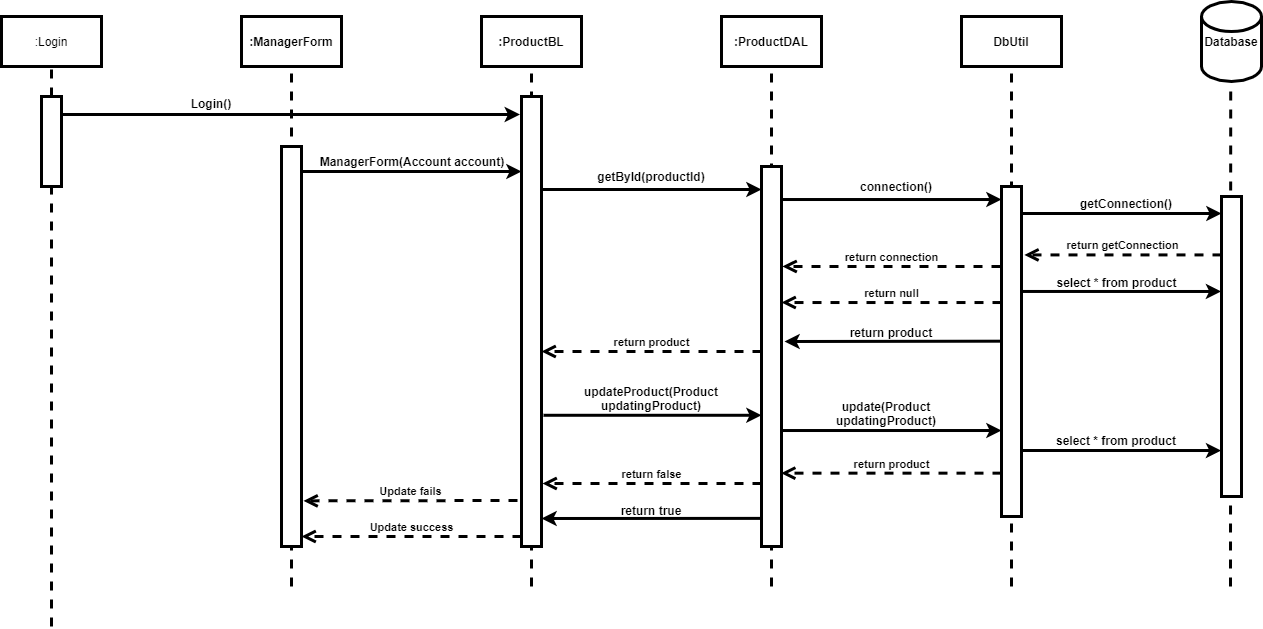
UC04\_Insert Product (Description)

|  |  |
| --- | --- |
| 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 | 1. Select *‘Thông tin sản phẩm’* tab  2. Select search method  3. Input key word to search  4. Press *‘Tìm 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 | AC05:   * Background color change to blue color (EX5)   AC7:   * Show confirm dialog * Press ‘yes’ button (EX07.1) * Display update information   AC07.2:   * Auto fill old information of that to corresponding fields   AC08:   * User can edit data on fields to update information   AC09:   * If user press *‘Hủy’* button   + Cancel update action  AC10:   * If user press *‘Hủy’ or ‘Cancel’* button   + Cancel update action  AC11:   * If update success   + Prompt: *‘Cập nhật thành công’*   * If update fails   + Prompt: *‘Cập nhật thất bại’* |
| Exceptions | EX05:  If background color wasn’t change to blue   * Prompt: *‘Vui lòng chọn sản phẩm’*   EX07.1:  If user press *‘no’* button  + Cancel update action |

UC04\_ Insert Product (Activity Diagram)



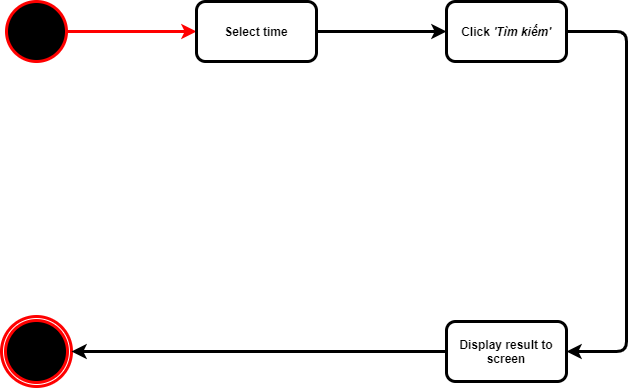
UC04\_Insert Product (SequenceDiagram)



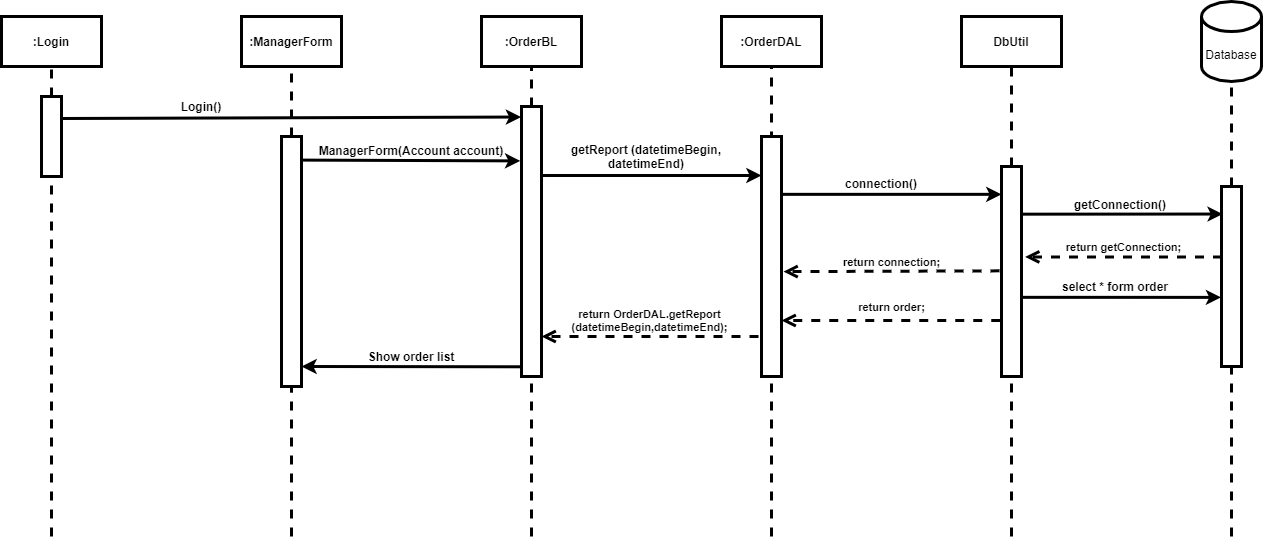
UC05\_Update Product (Description)

|  |  |
| --- | --- |
| 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 | 1. Select time (AC1) 2. Click *‘Tìm kiếm’* 3. Display results to screen (AC3) |
| Alternate Courses | AC1: Select time   1. In field begin day   + Set day, month and year   1. In field End day   2.1: Set day, month and year  2.2: User press check box to get real time  AC3:  1. User select order to see details (See UC04\_AC05) |
| Exception | EX2.2:   * If user press check box to get real time   + Not allow to user choose end date time themself |

UC05\_Update Product (Activity Diagram)



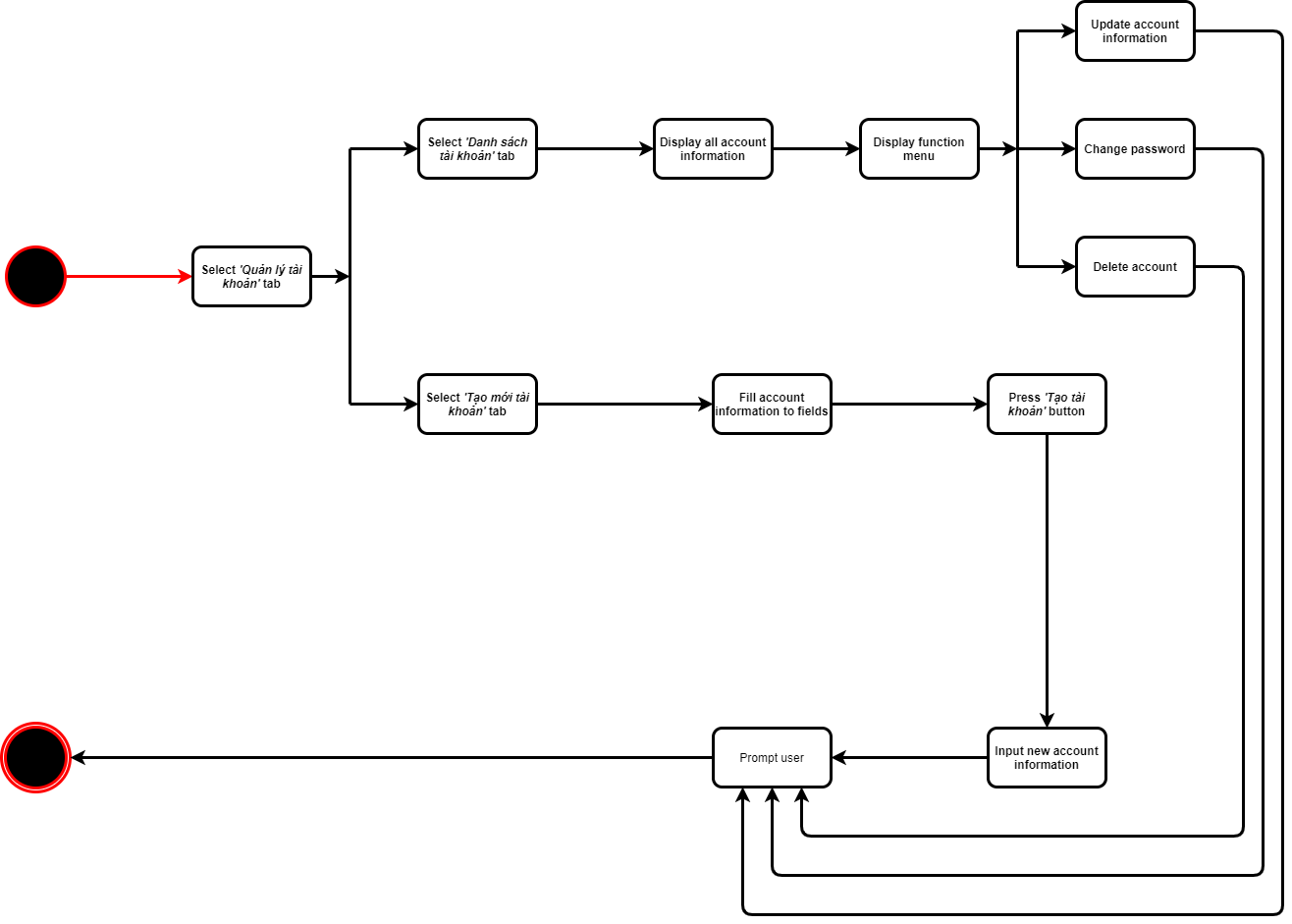
UC05\_Update Order(SequenceDiagram)



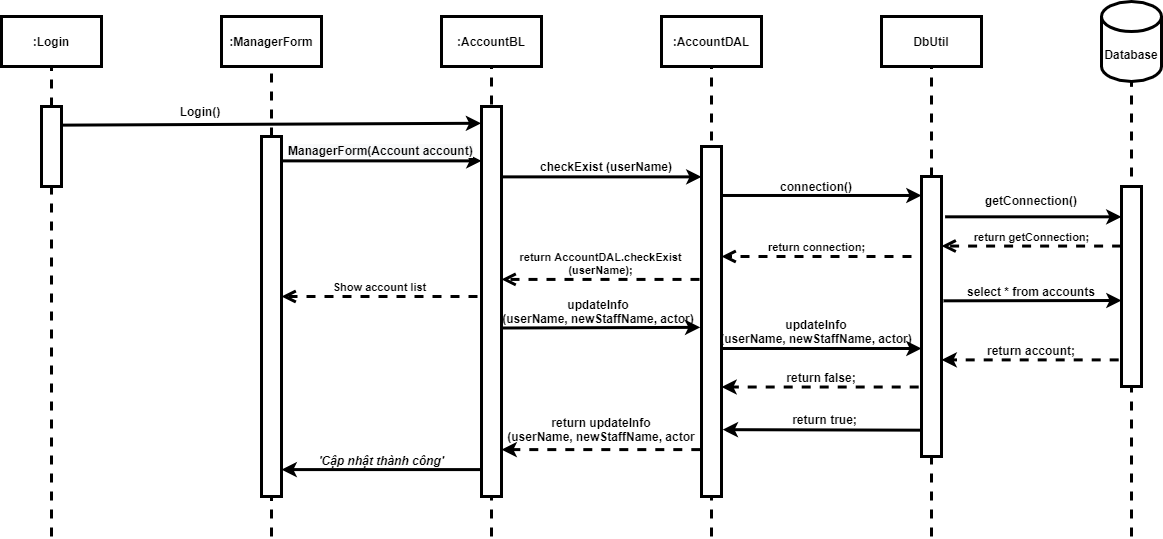
UC06\_ Management Account (Description)

|  |  |
| --- | --- |
| 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 | + *‘Danh sách tài khoản’* tab (AC1) (AC1.1)(AC1.1.2)  + *‘Thêm mới tài khoản’* tab (AC2) |
| Alternate Courses | AC1:   1. Display all accounts information to screen   AC1.1: Display function menu  1. See UC04\_AC05  AC1.1.1: Update accounts information  1. Display update dialog  2. Fill information to corresponding fields  3. Press *‘Cập nhật’* button  AC1.1.2: Change Password   1. Display confirm dialog (EX1.1.2.1)   AC1.1.3: Delete account   1. Display confirm dialog   AC2: Select *‘Thêm mới tài khoản’* tab   1. Fill account information to fields 2. Fress *‘Tạo tài khoản’* button (EX2.2) |
| Exception | EX1.1.1.3:   * User press *‘Hủy’* button   + Cancel update action  EX1.1.2.1:   * User press *‘No’ or ‘Cancel’* button   + Cancel action   * User press *‘Yes’* button   + Display input dialog  If account changing is staff account   * User input new password * Confirm   If account changing is manager account   * User input old password   + Valid it   * User input new password * Confirm   EX1.1.3.1:   * User press *‘No’ or ‘Cancel’* button   + Cancel action   * User press *‘Yes’* button   + Delete account  + Prompt result action  EX2.2:   * User press *‘Hủy’* button   + Clear all fields  + Cancel action |

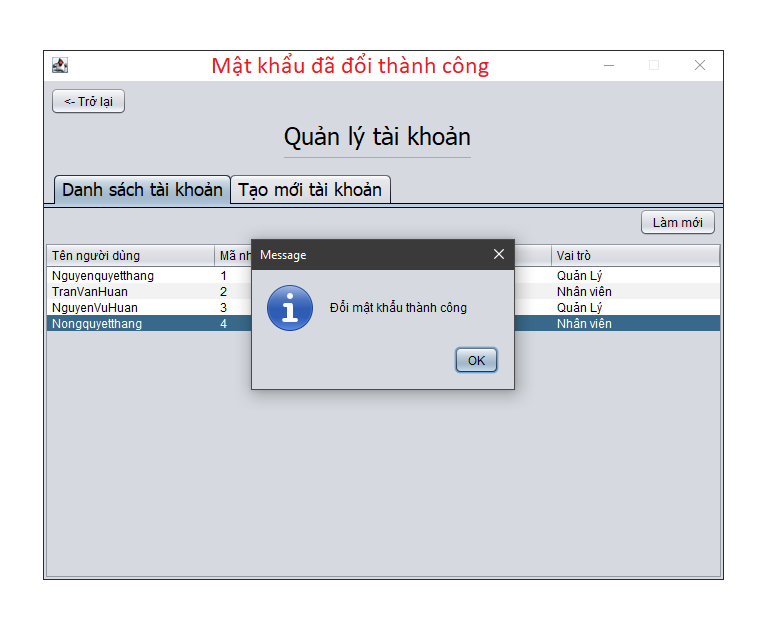
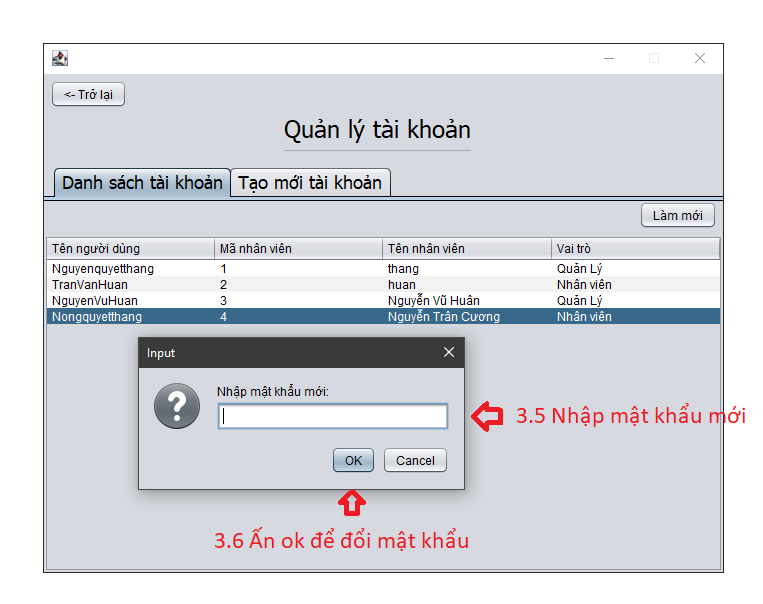
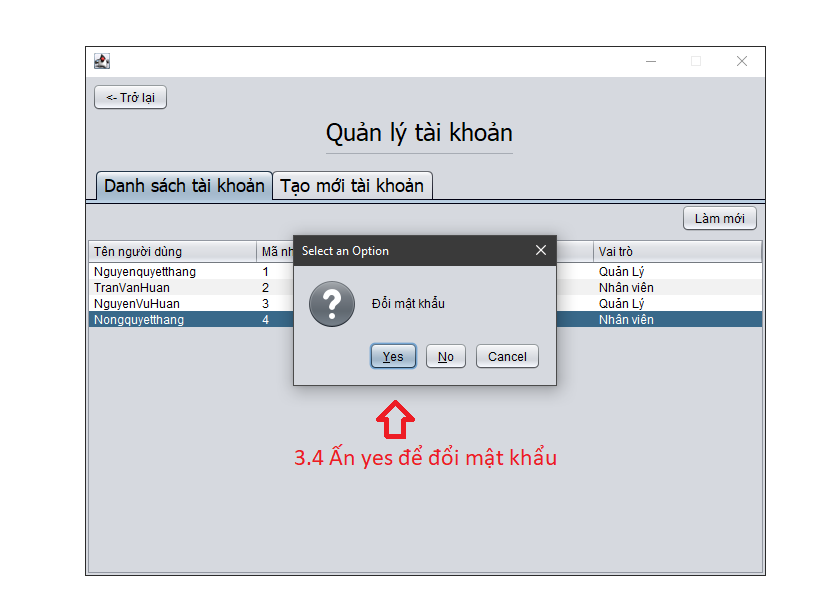
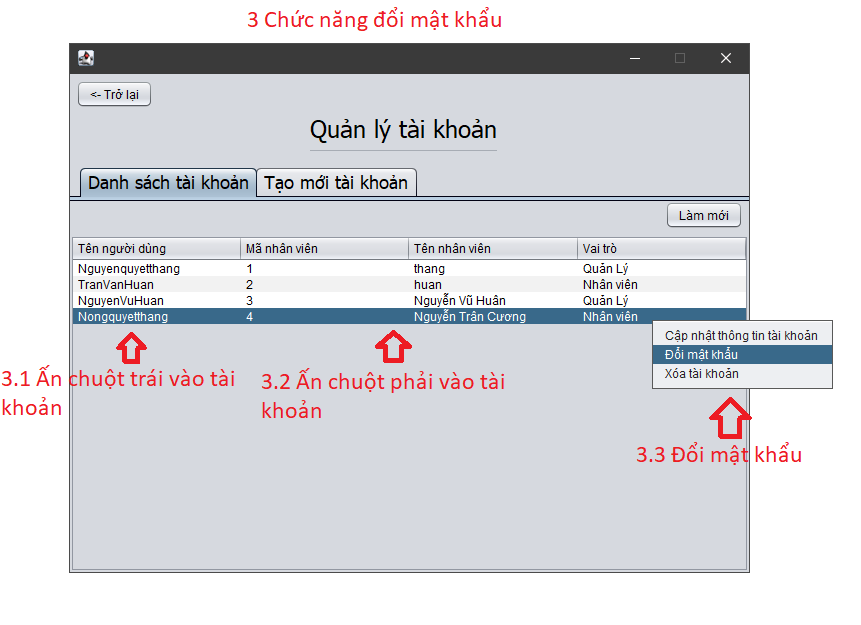
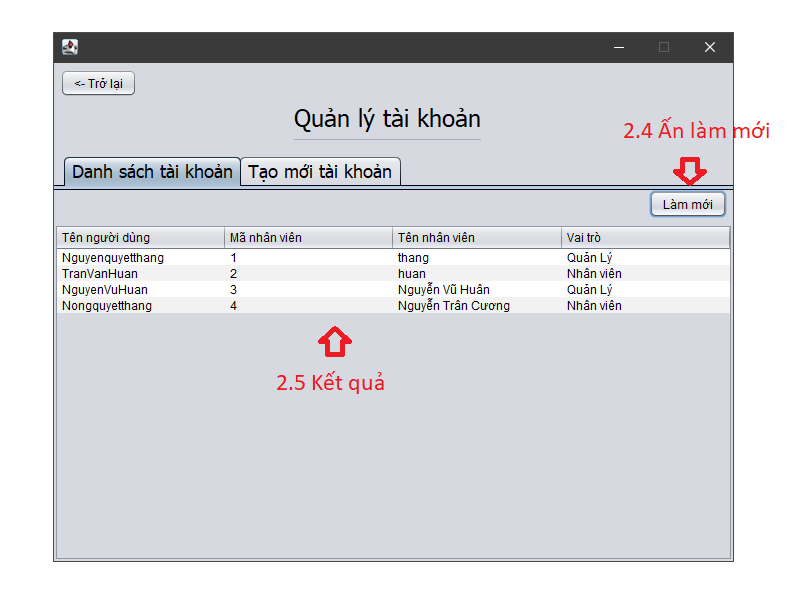
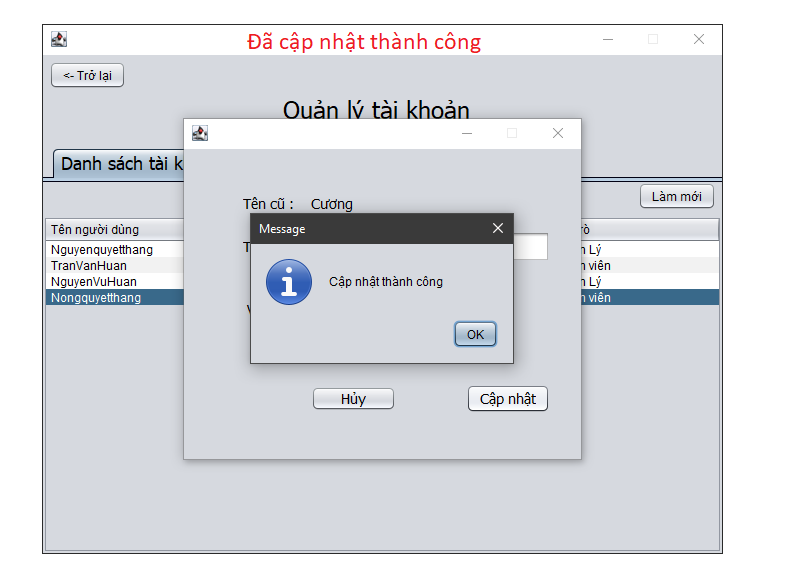
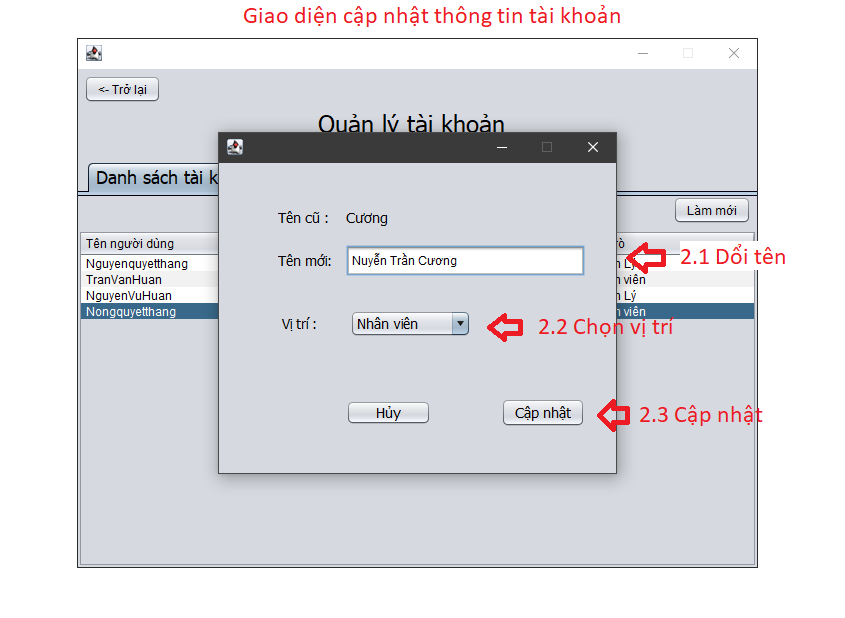
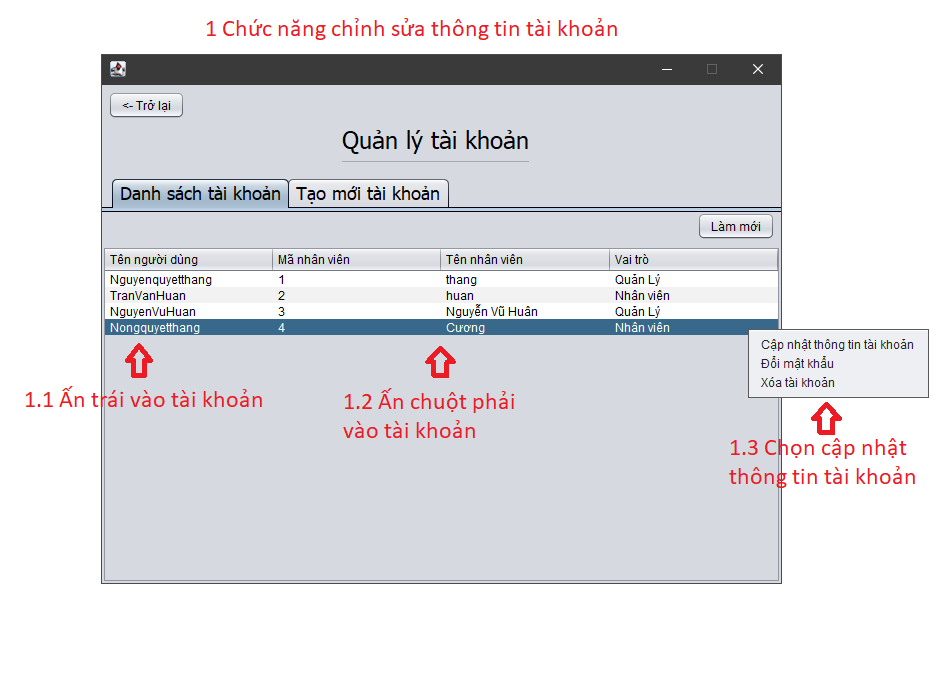
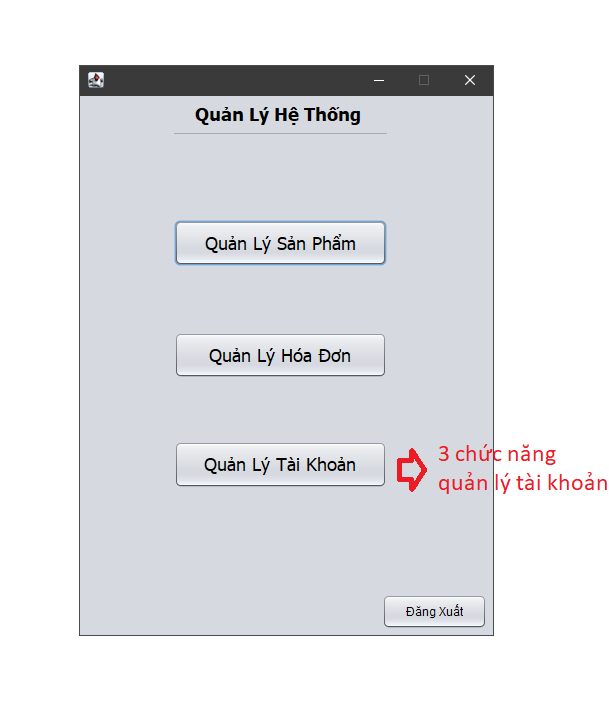
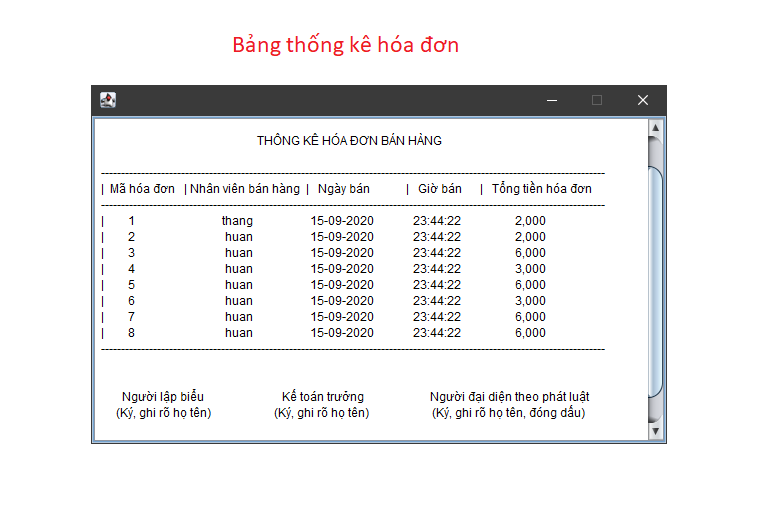
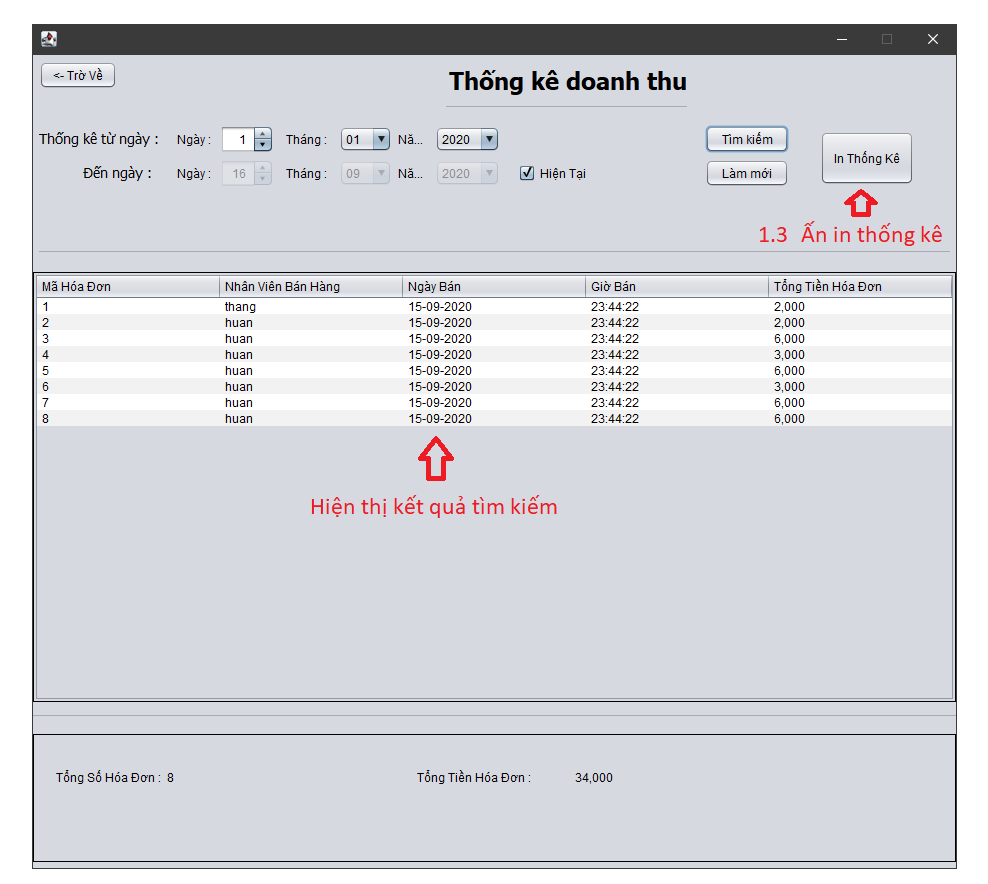
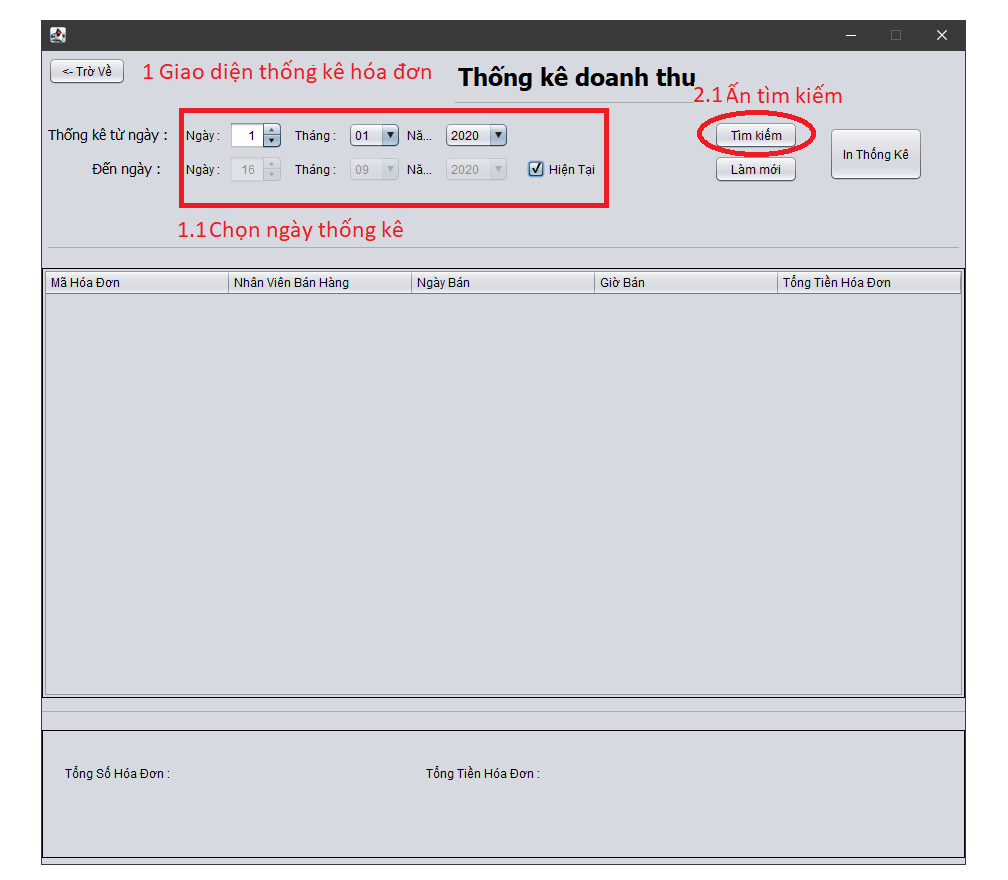
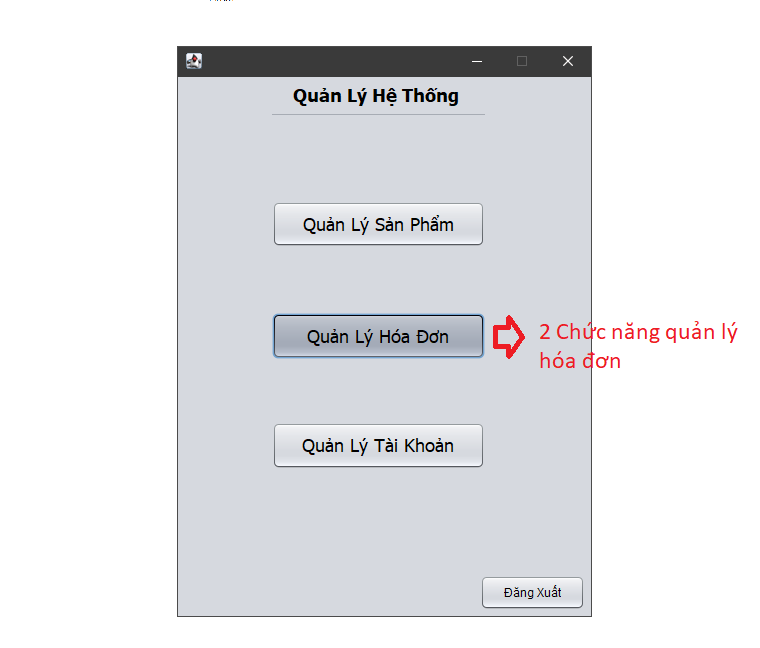
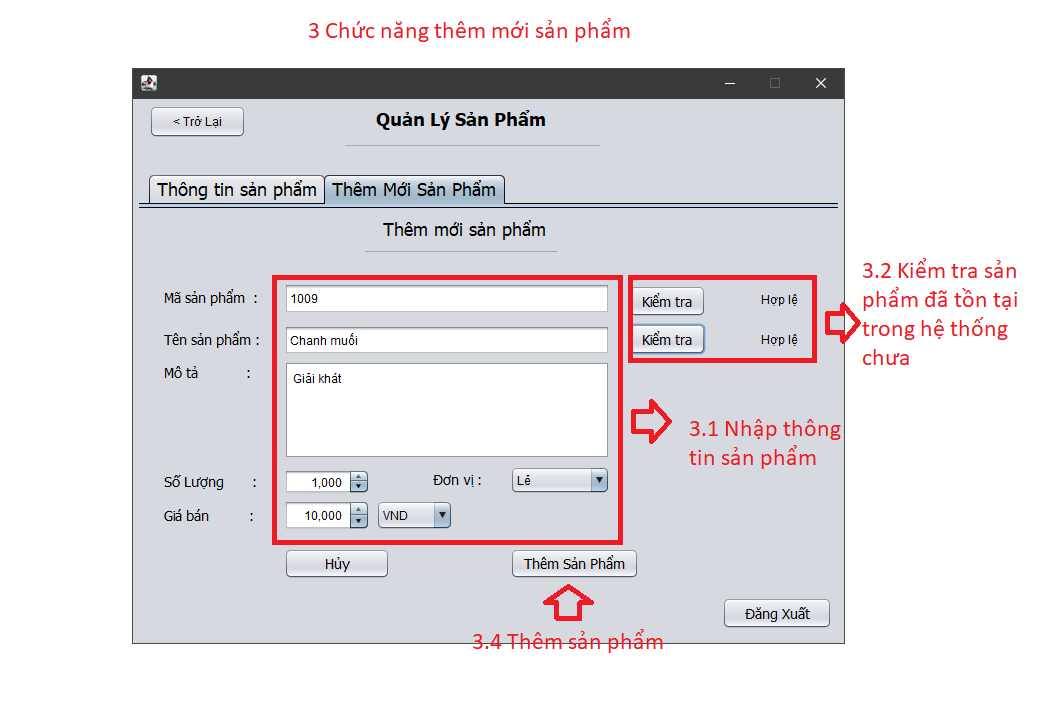
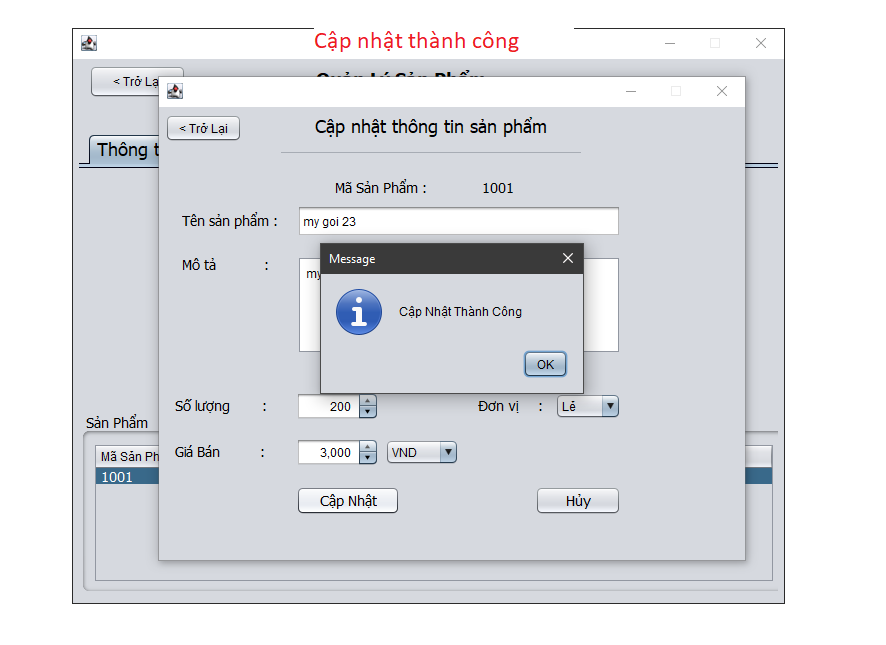
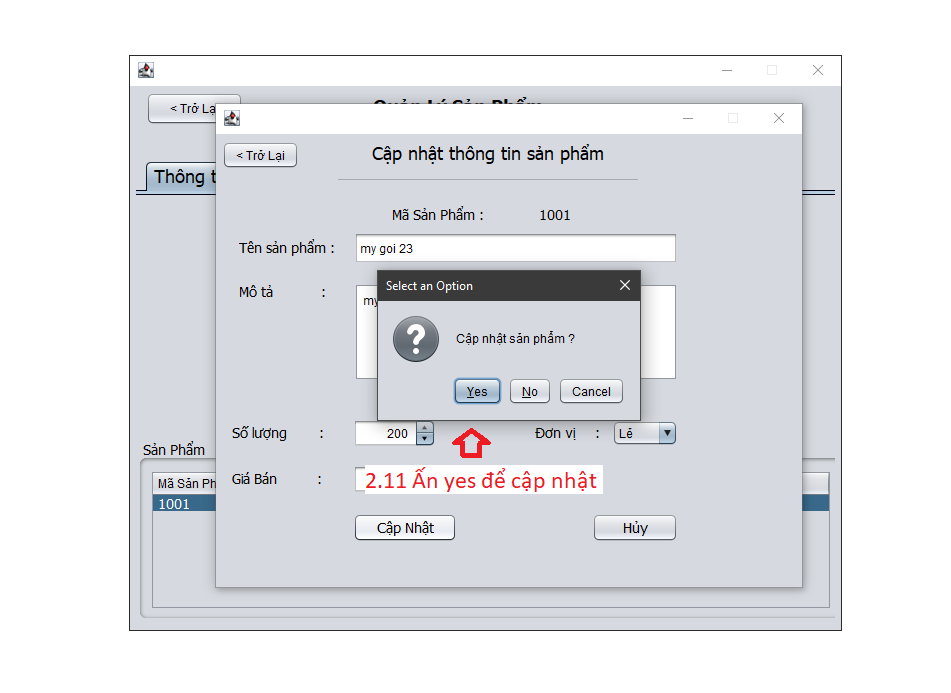
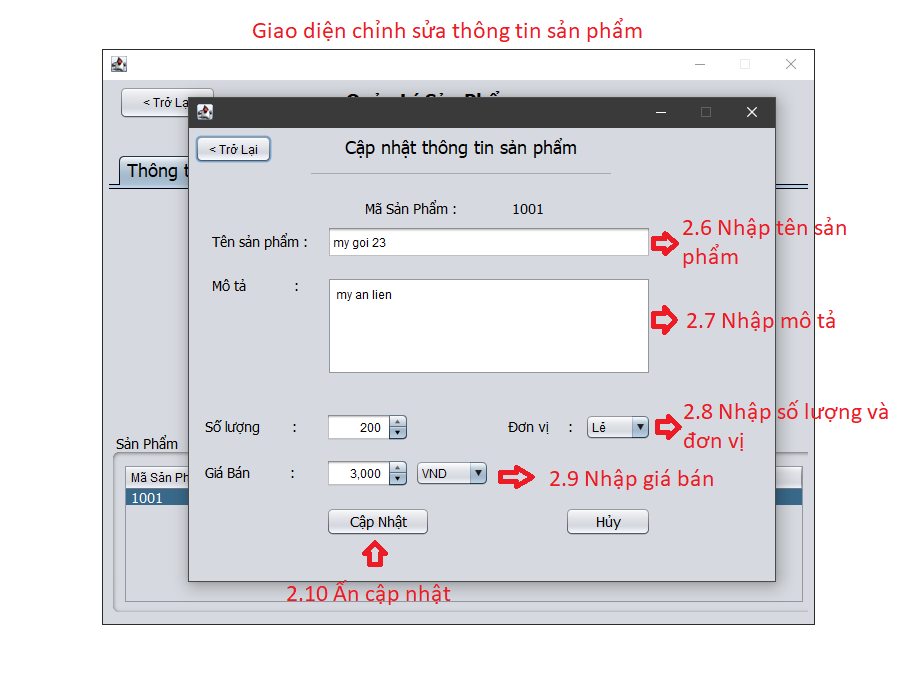
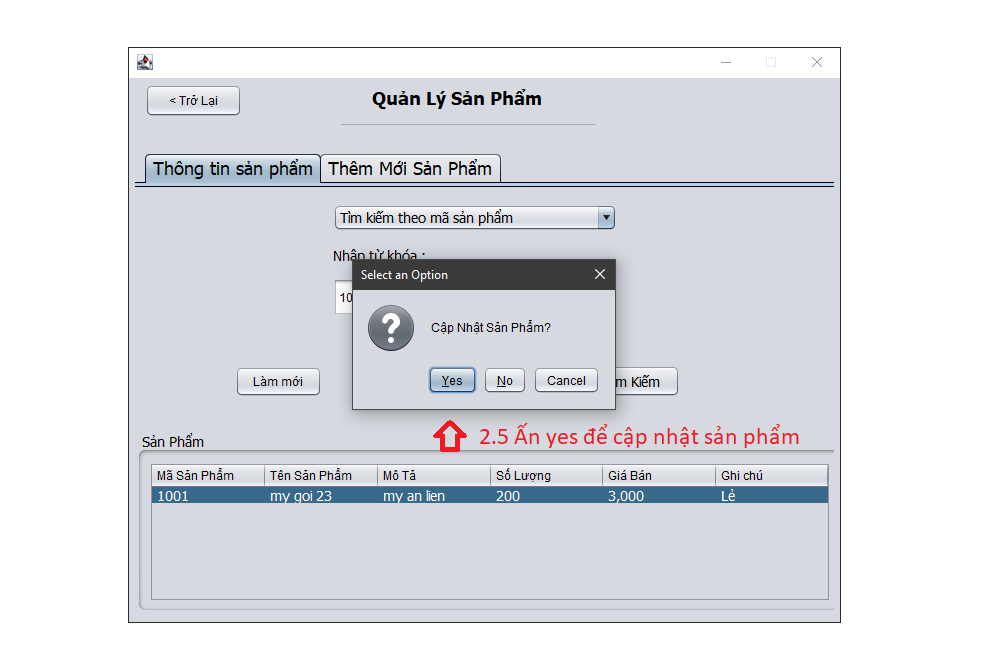
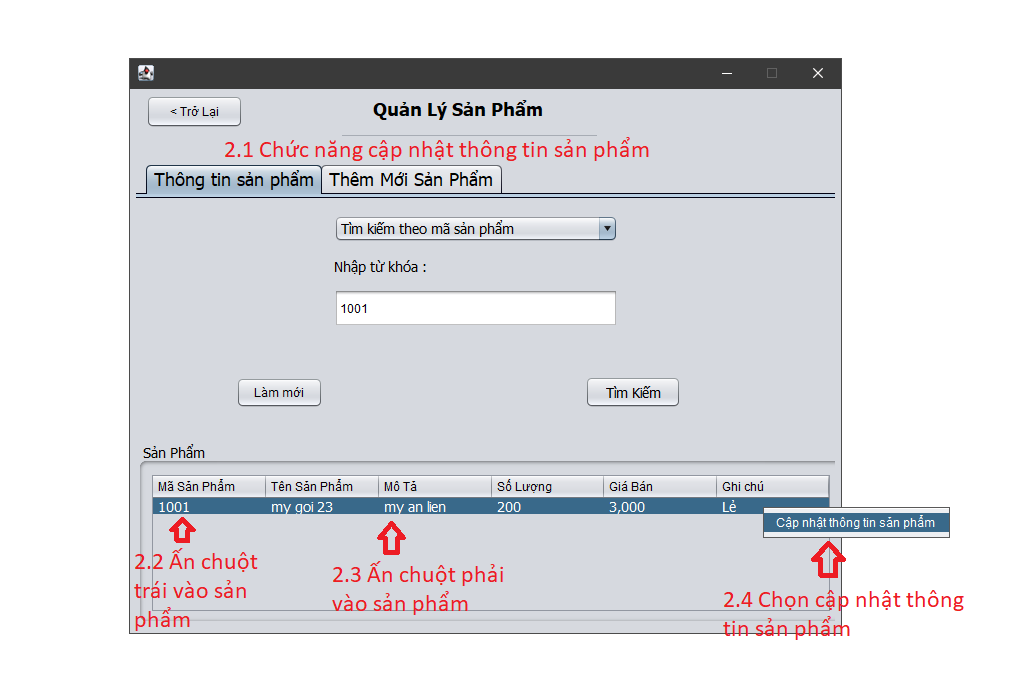
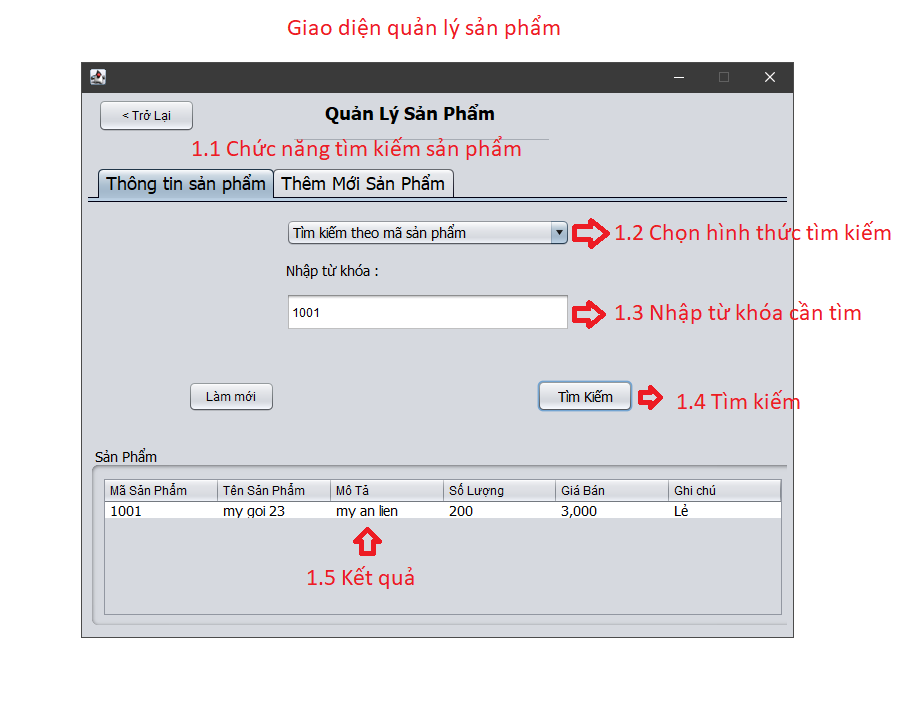
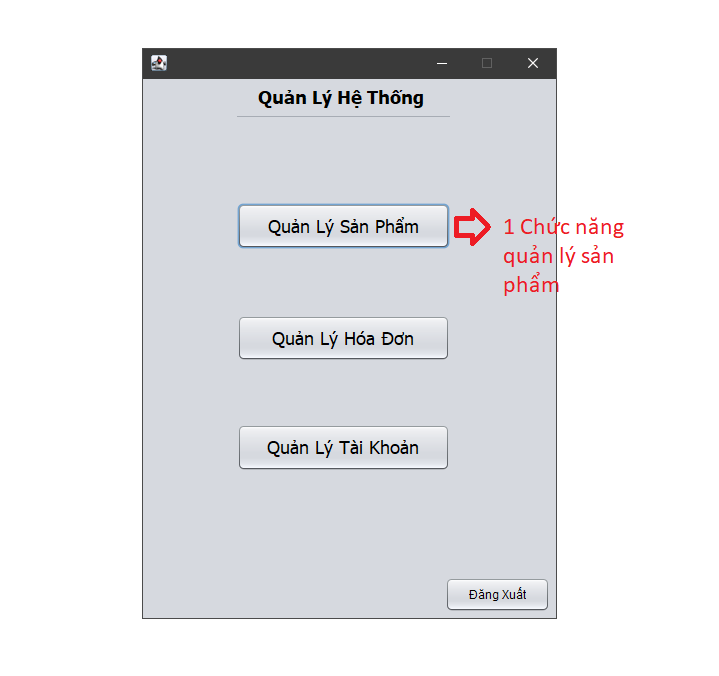
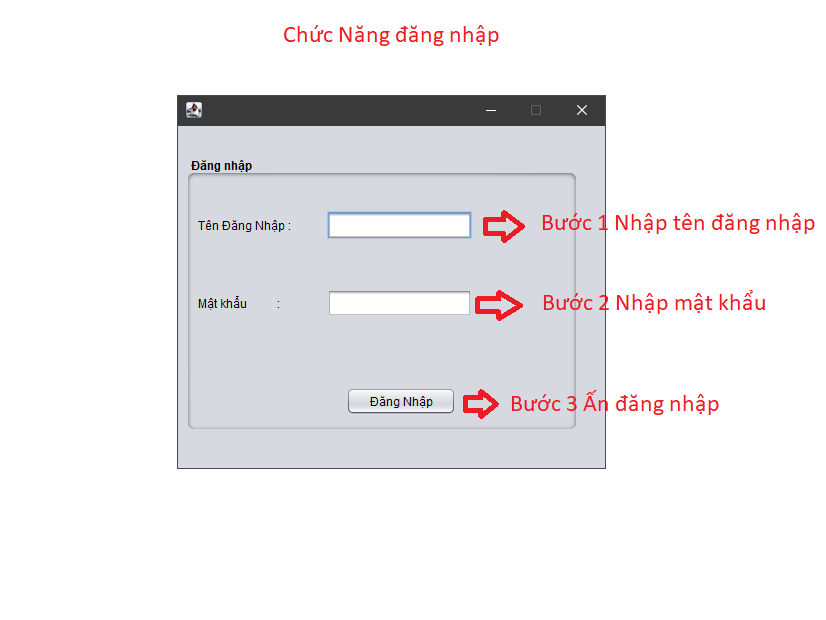
UC06\_ Management Account (Activity Diagram)

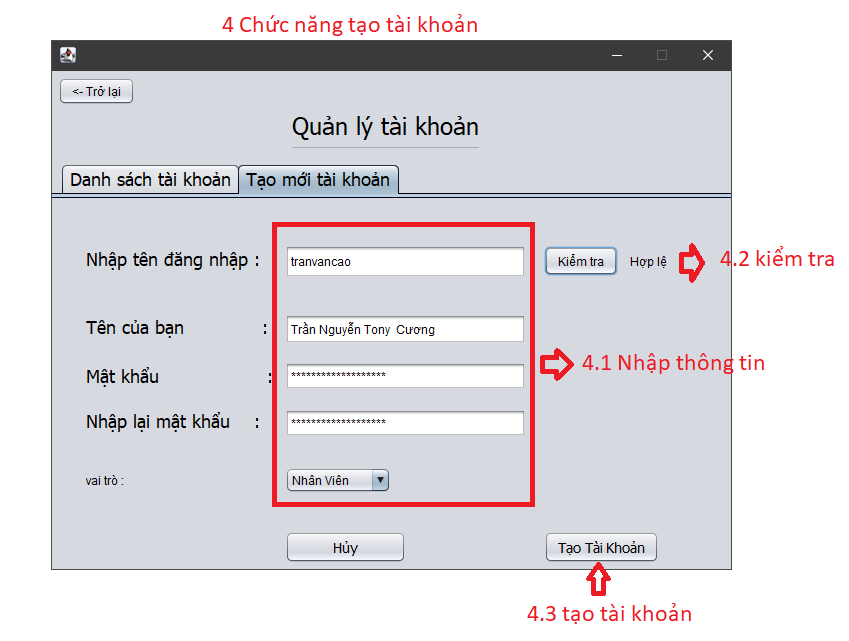


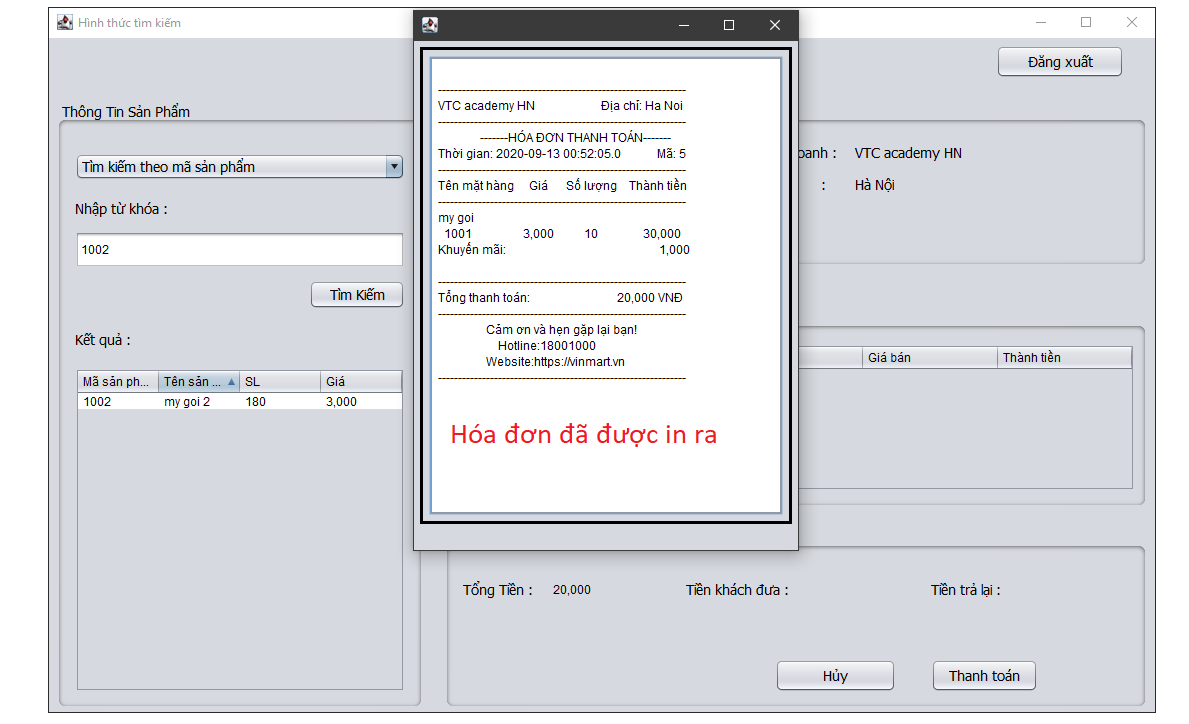
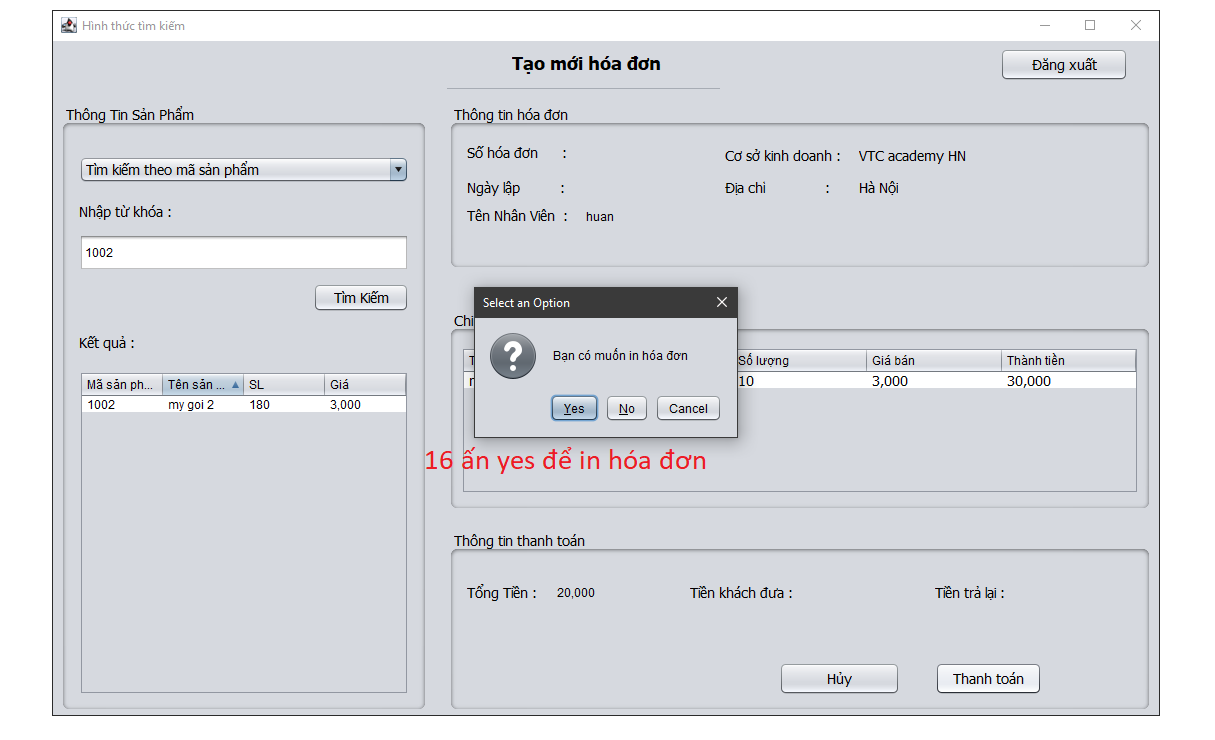
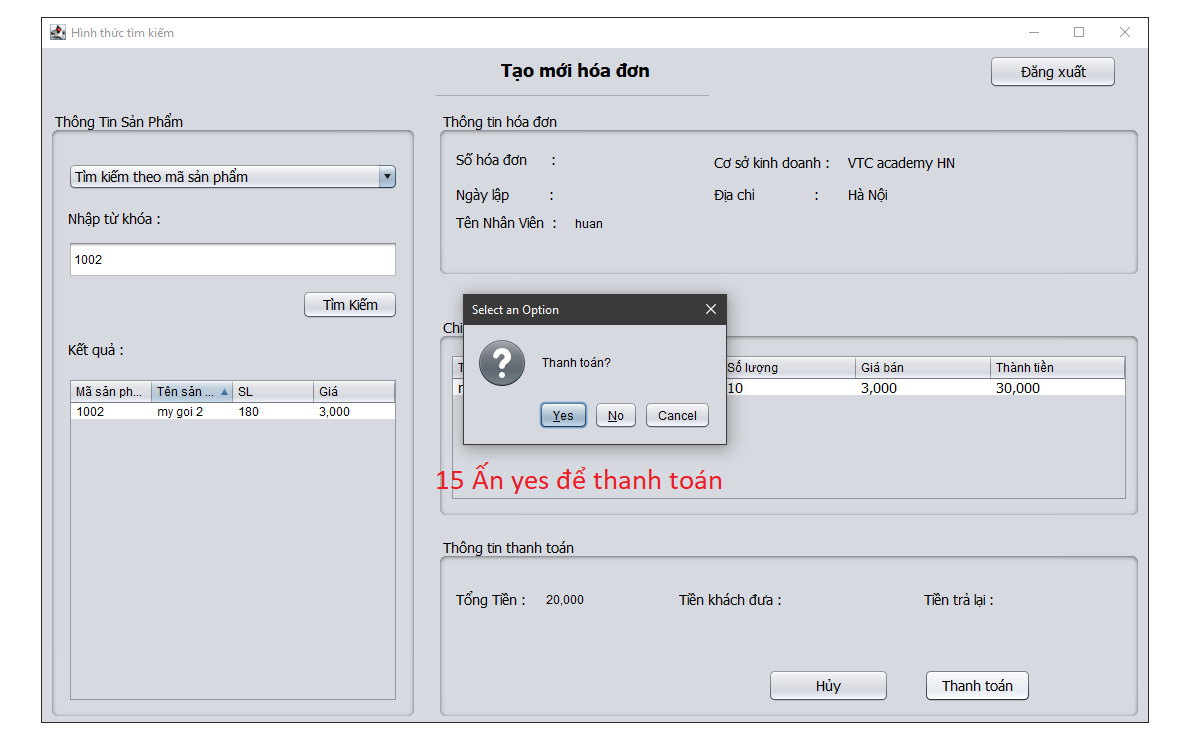
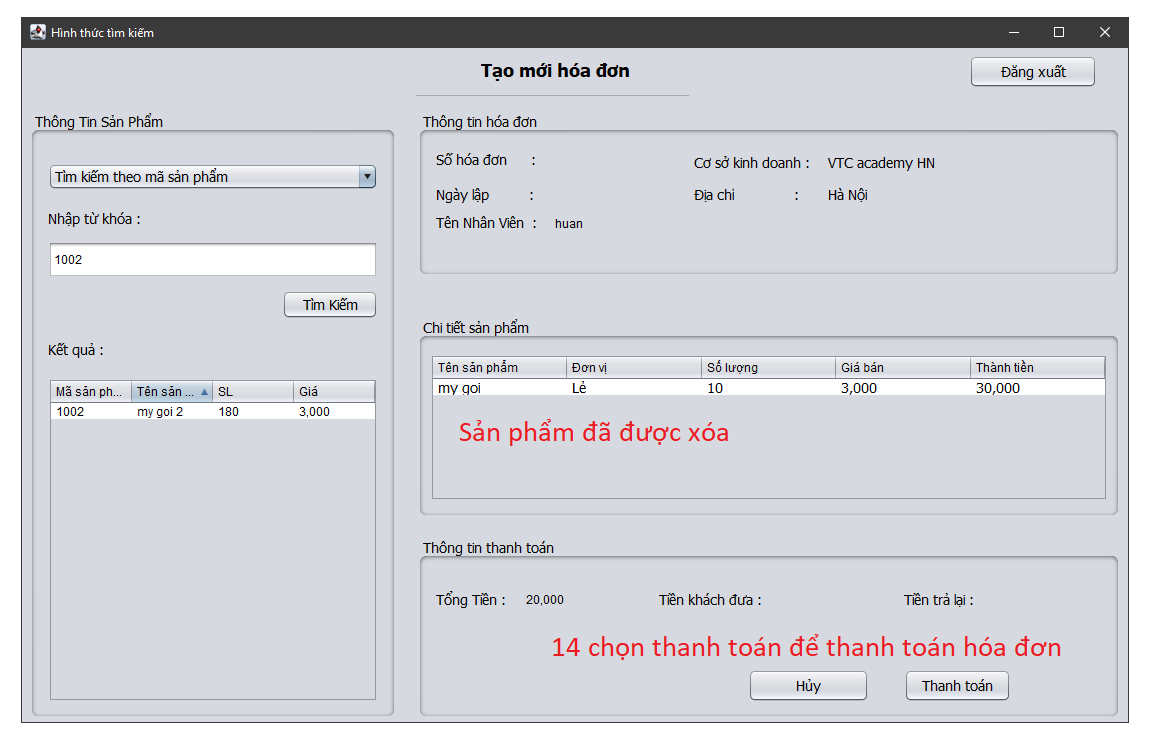
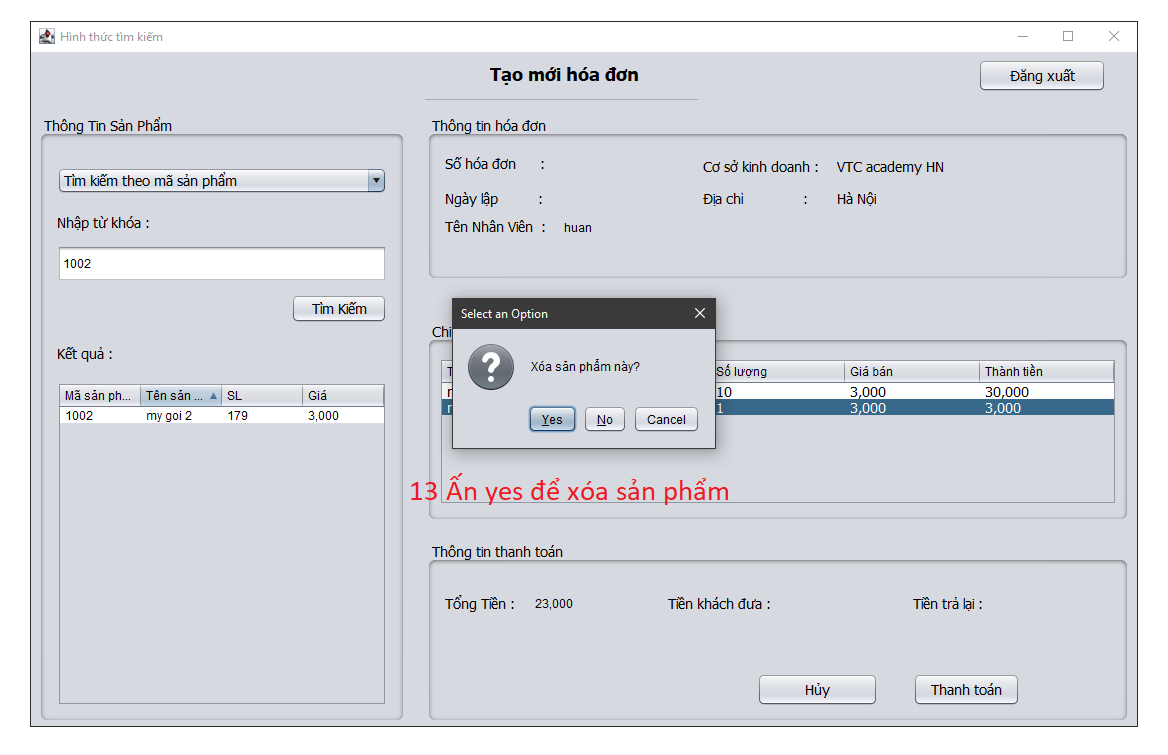
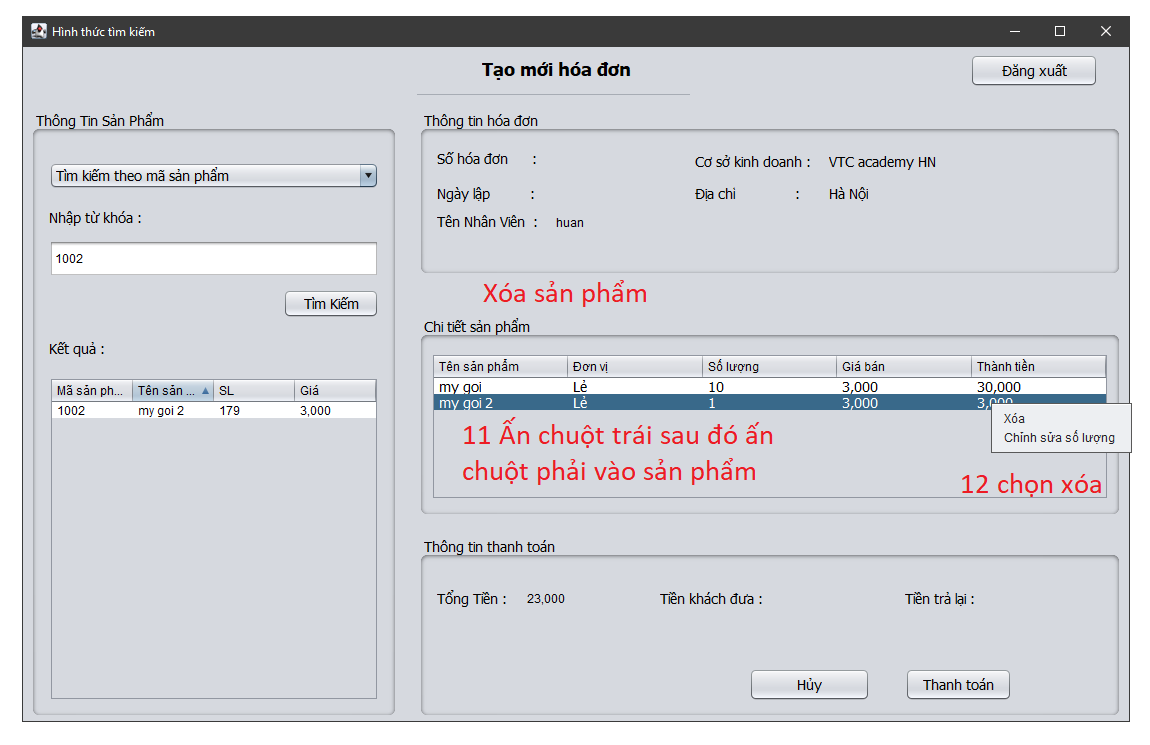
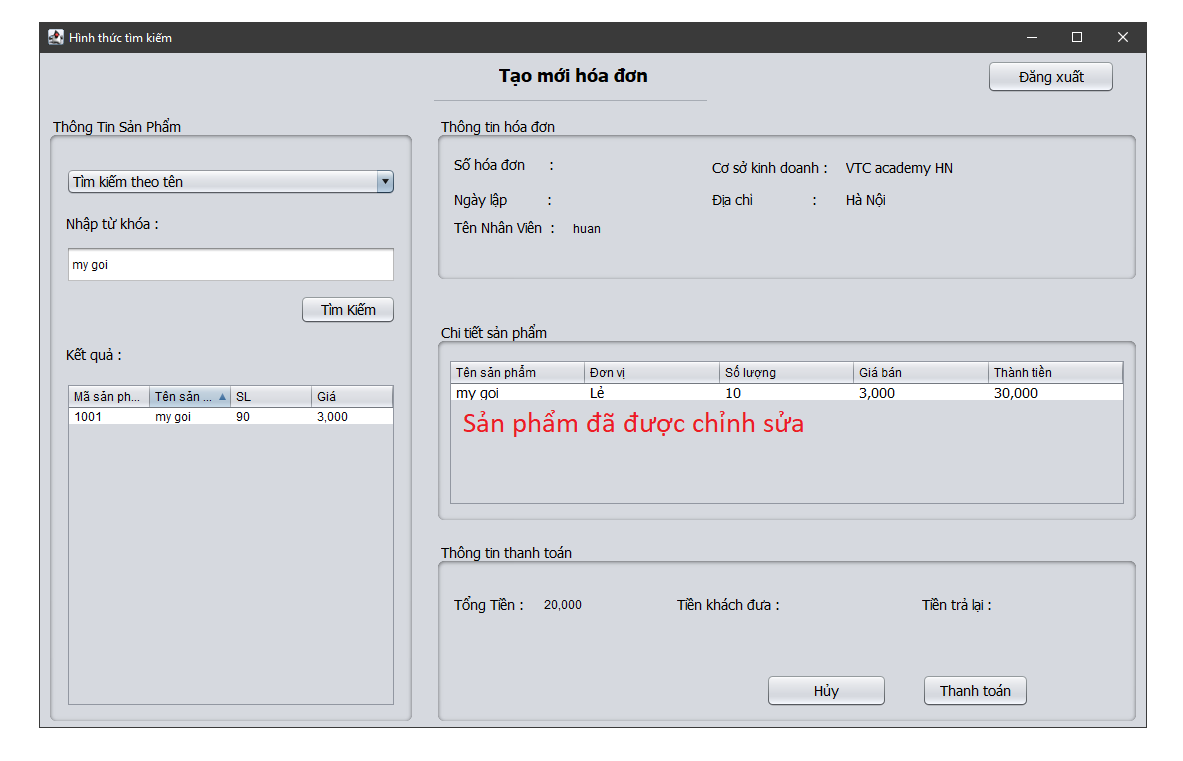
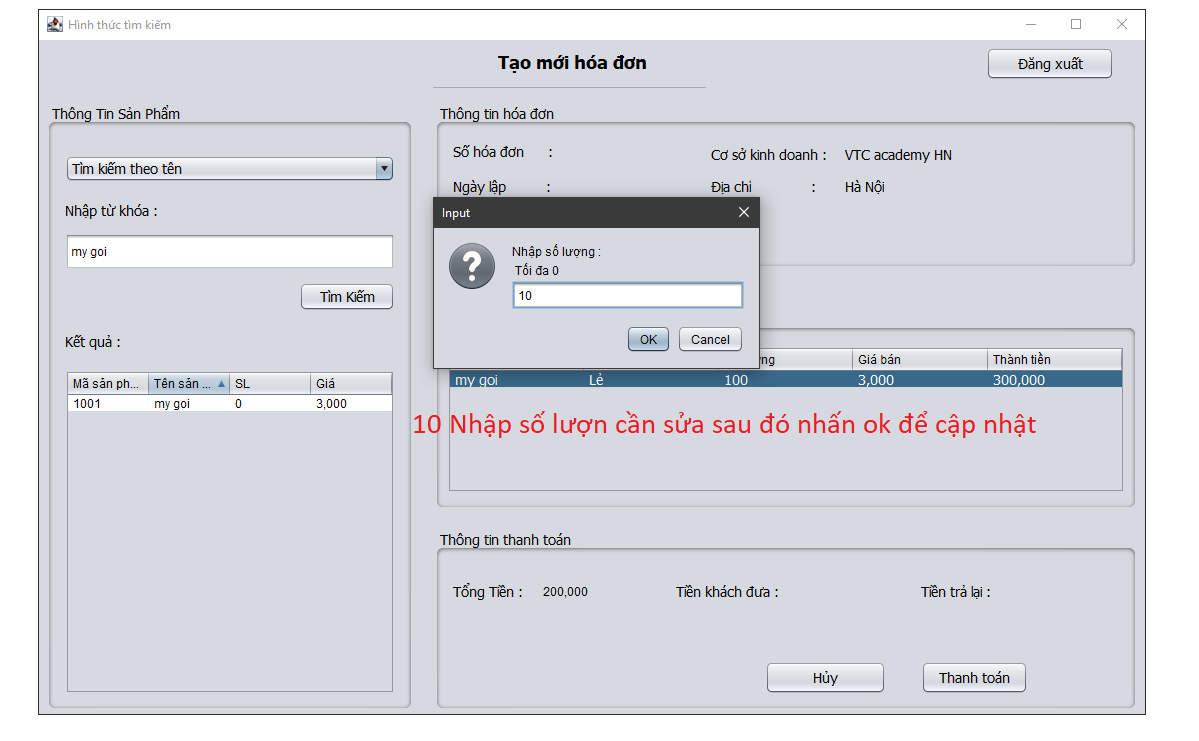
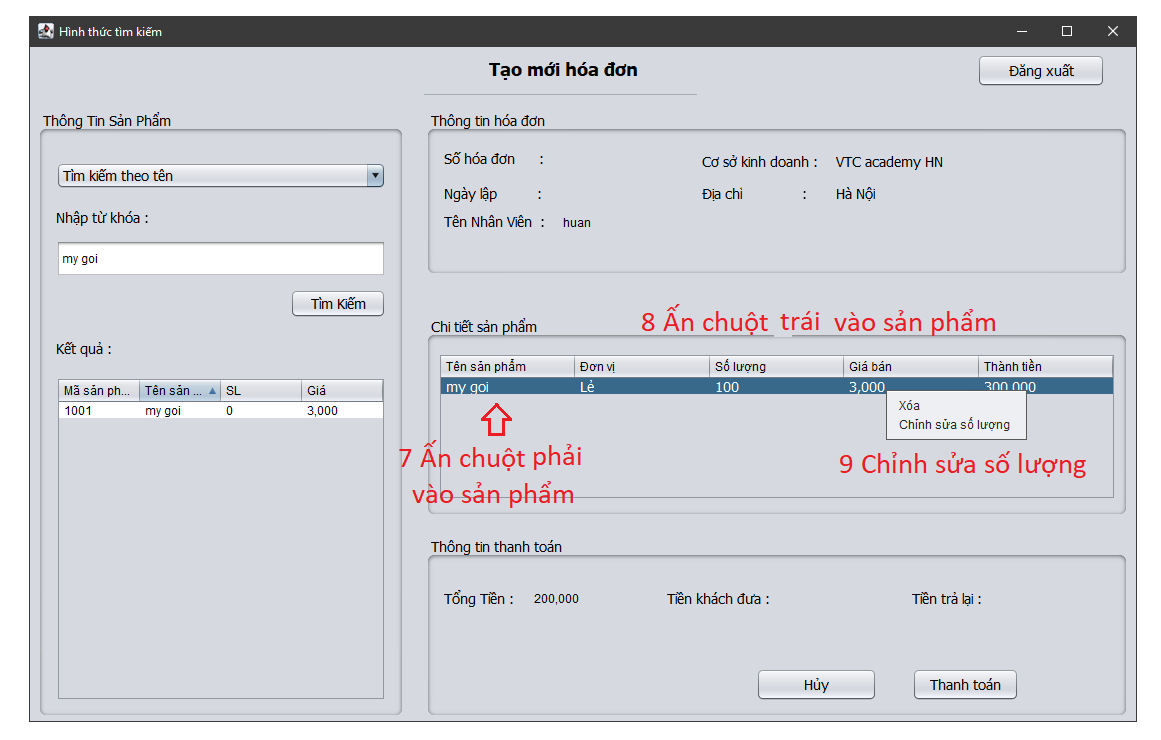
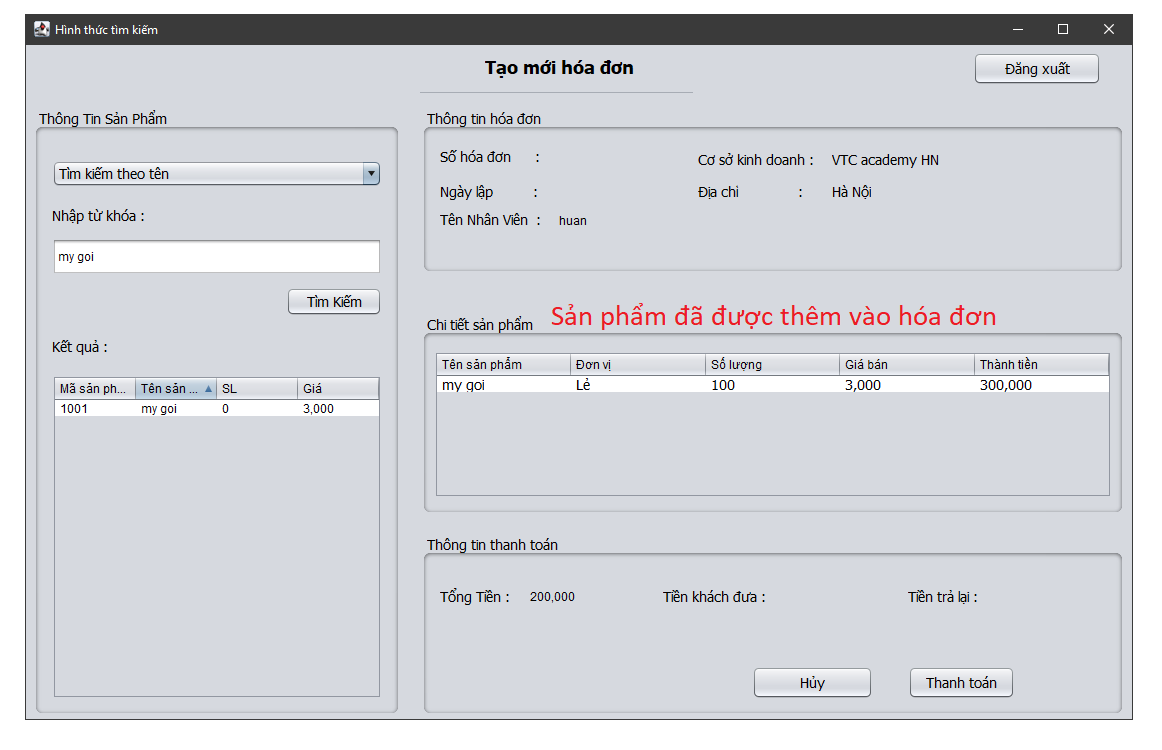
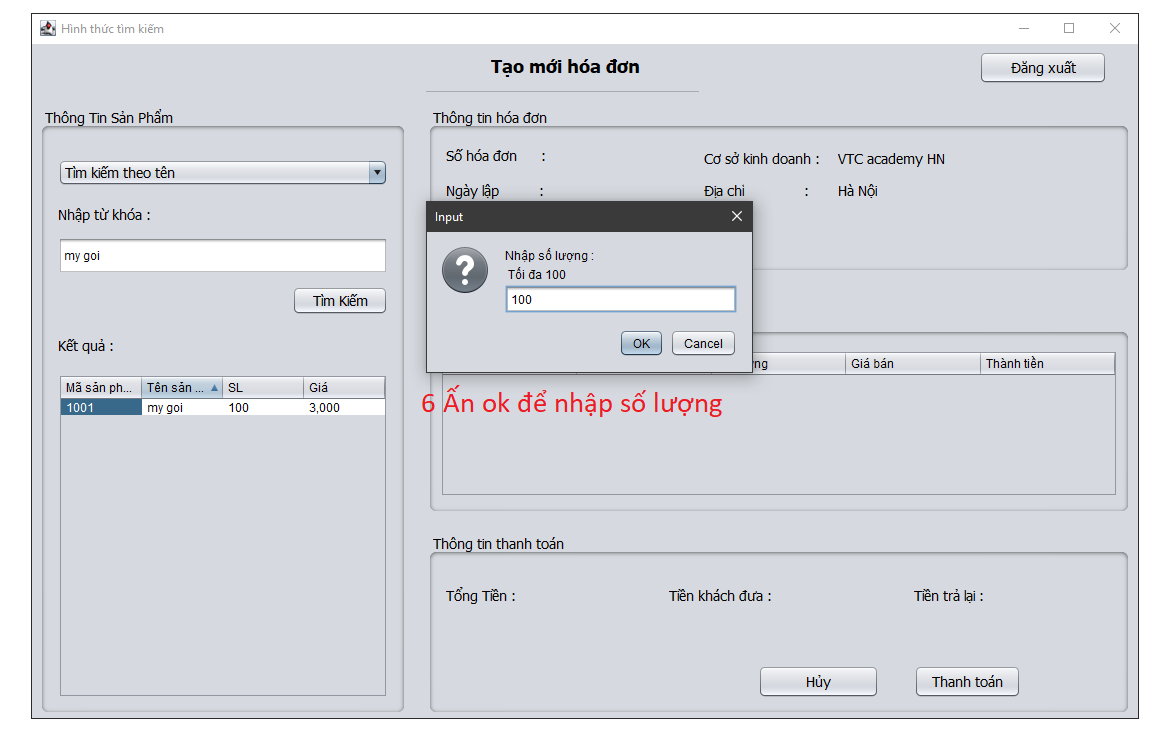
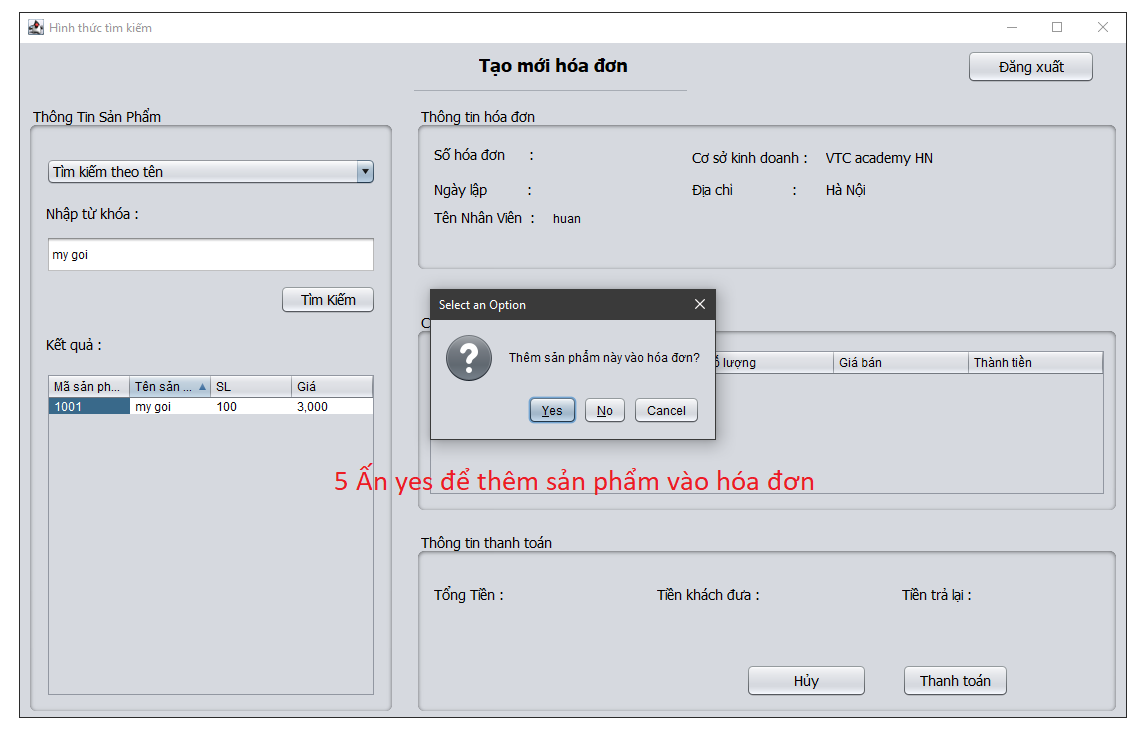
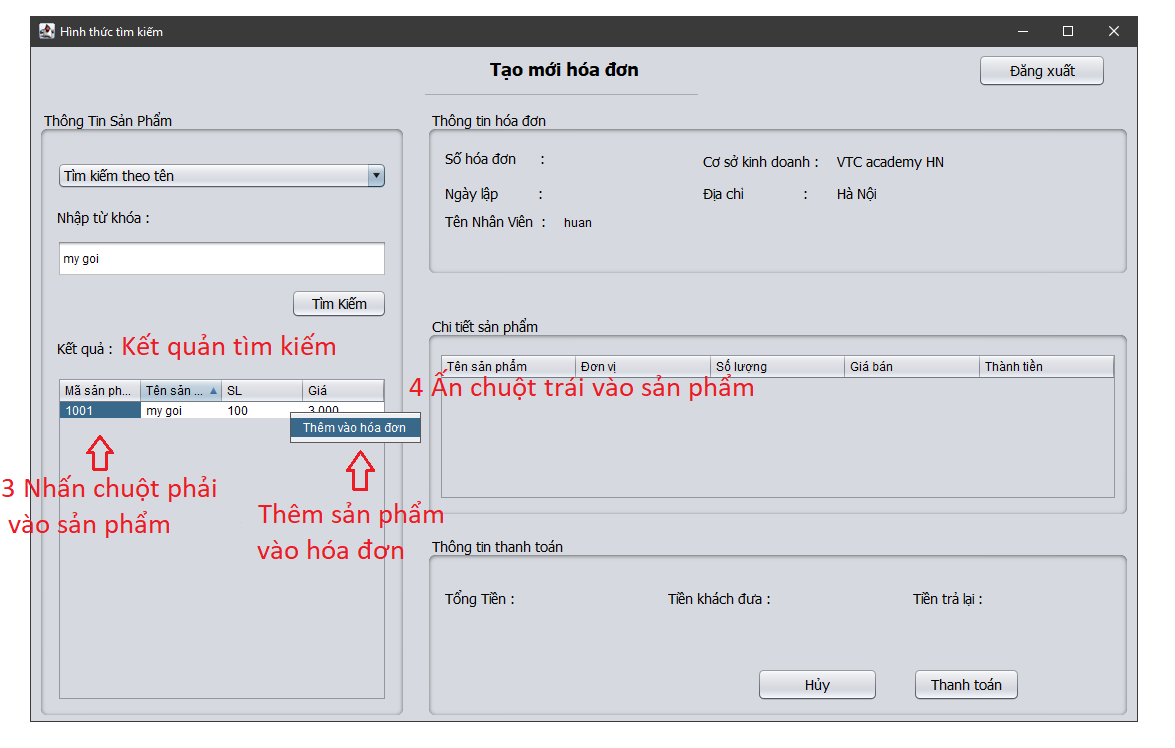
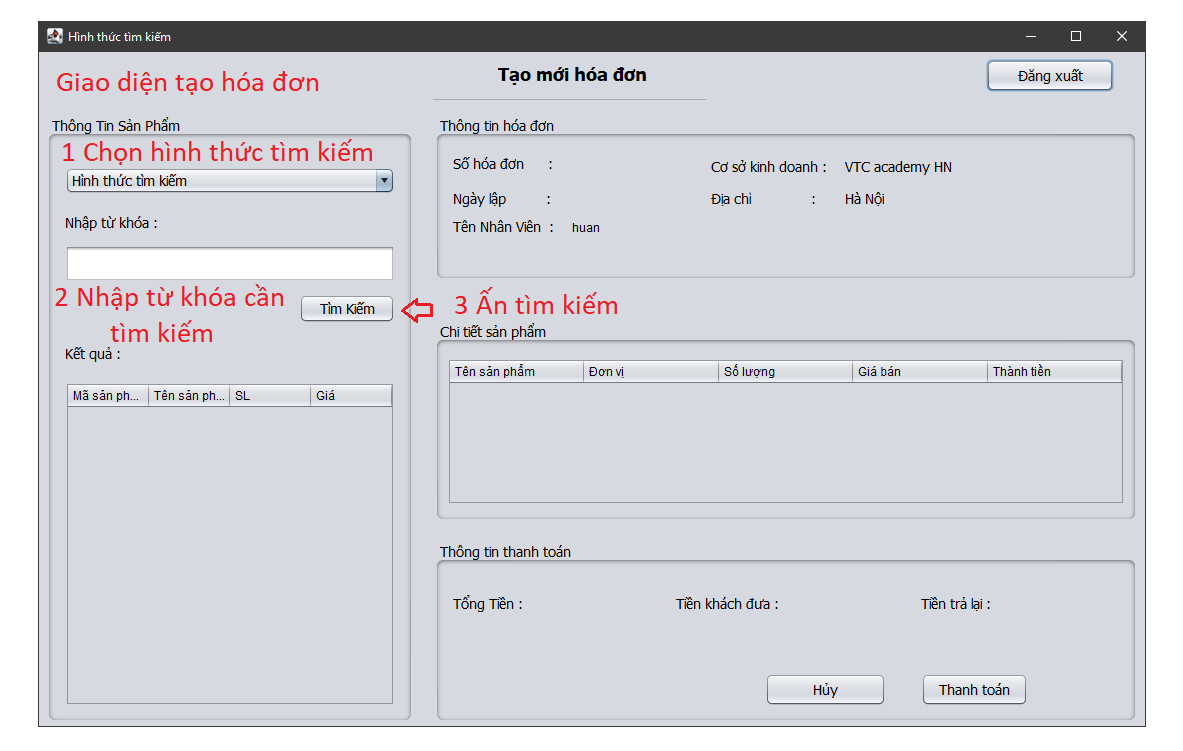
UC06\_ Management Account (SequenceDiagram)



III. Design Details

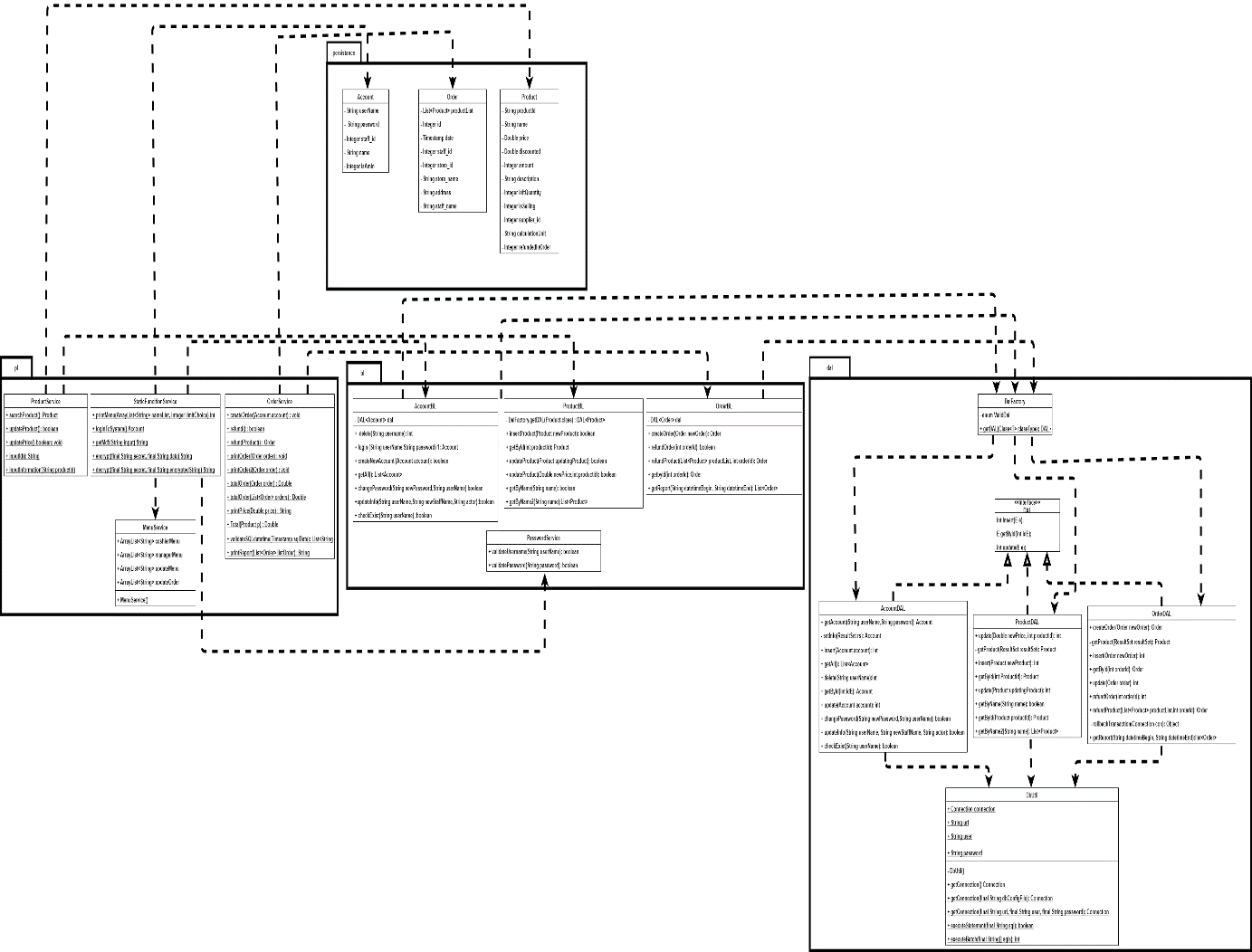
1. Manager



2. Cashier

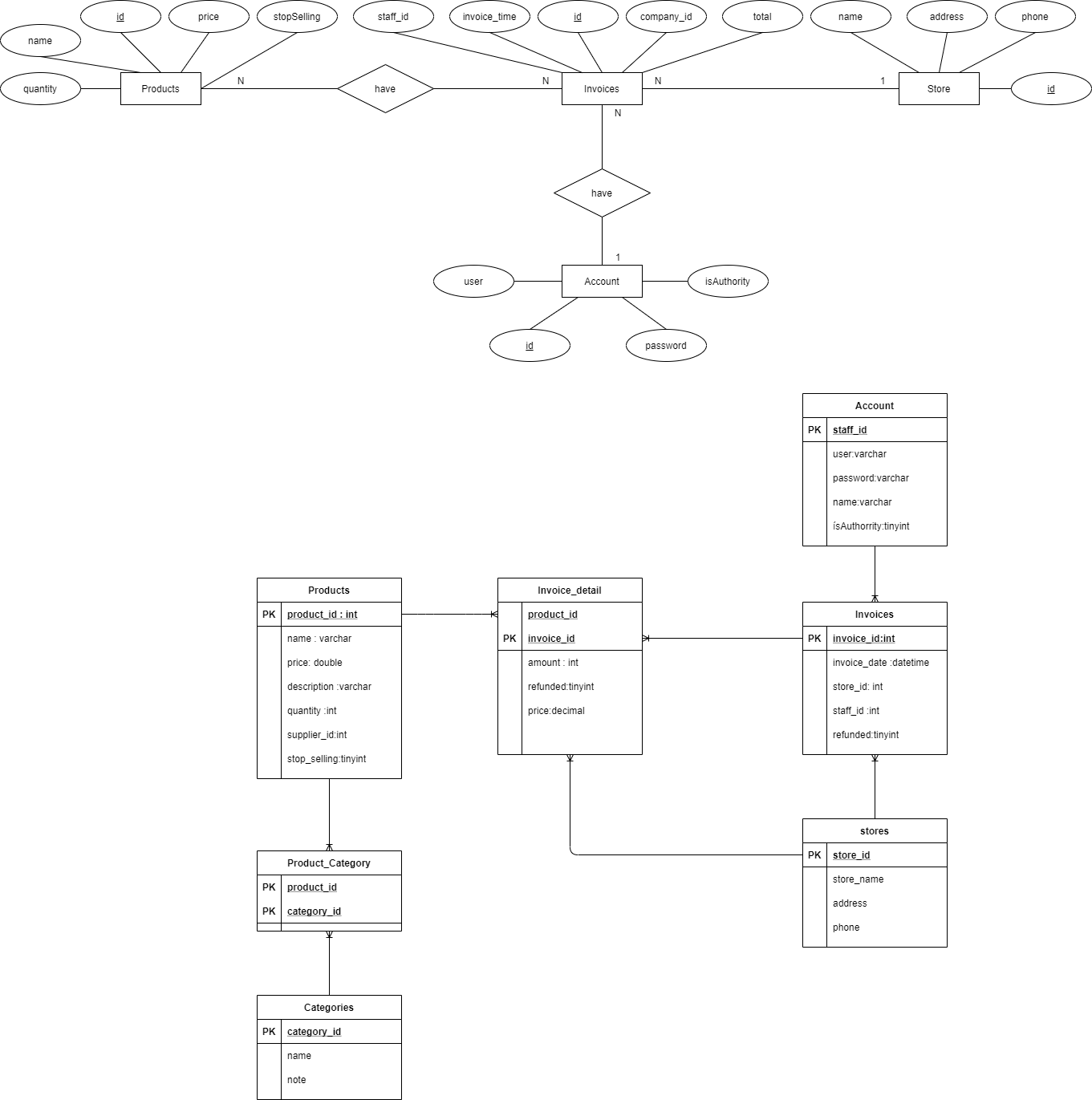
2. Code Design (Class Diagram)

(Class Diagram):

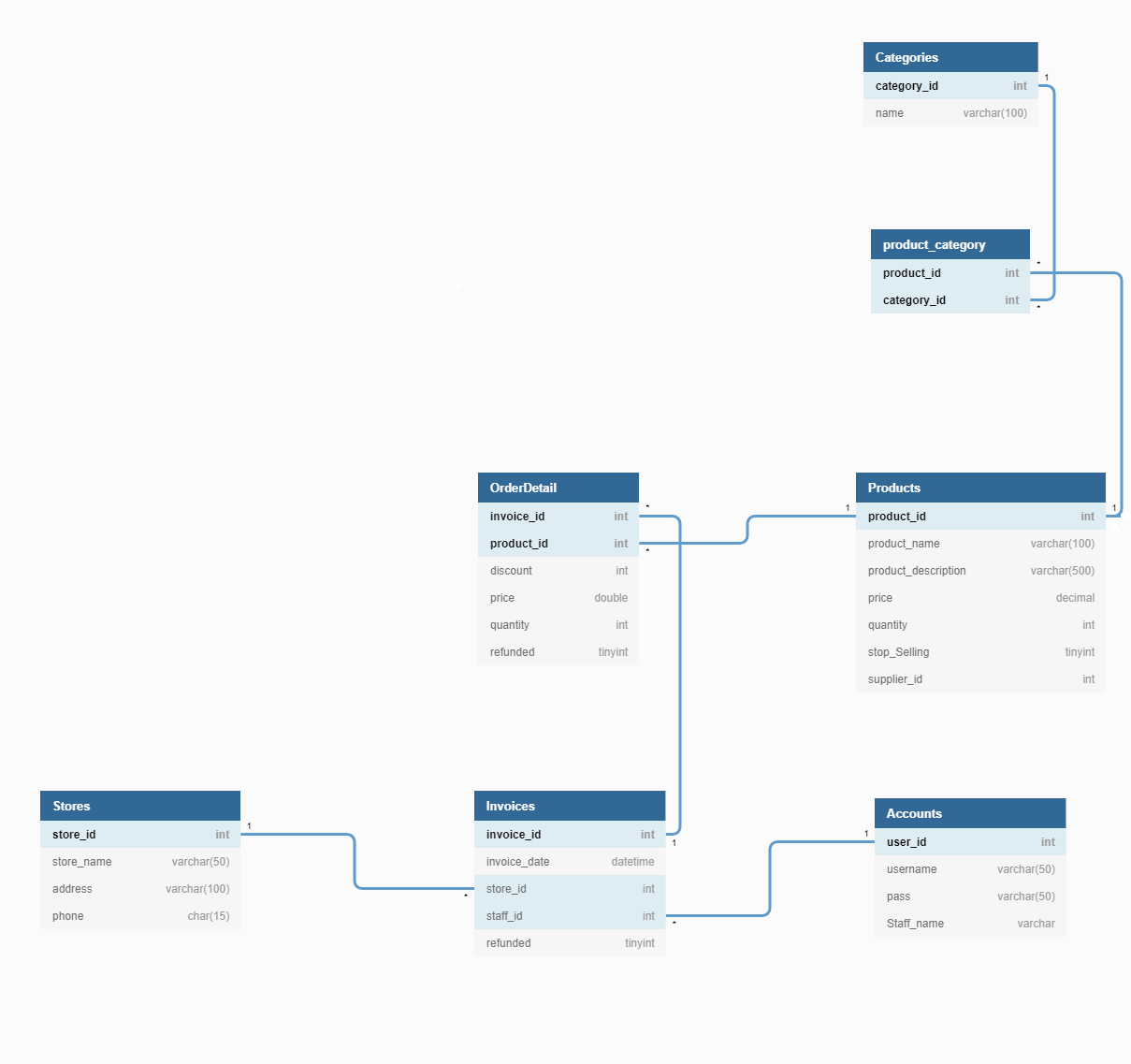


3. Database Design

a. Entity Relationship Diagram



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","16c0ce36e334e22fda8caca1  b10c2f9c", "Nguyenquyetthang",null,1,"thang",0 |
| Test Case expected output | True |
| Test Case Steps | Step1: create instance of acccountDAL  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 acccountDAL  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 acccountDAL  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 acccountDAL  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 | UserName mustn't have special charater 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 | UserName more than 20 character |
| Preconditions |  |
| Test Case Input | “Nguyenquyetthang123asdvghasdvghasvghasvdhvas” |
| Test Case expected output | False |
| Test Case Steps | Step1: Call validateUsername()  Step2: Input value  Step3: Test |
| Default Value preverving |  |

|  |  |
| --- | --- |
| Test CaseNumber | 1.9(validdatePassword) |
| 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 (validdatePassword2) |
| 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(validdatePassword3) |
| Test Case Name | Test password |
| Test Case description | It wans’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 |  |

1. 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 |  |

1. Test product

|  |  |
| --- | --- |
| Test CaseNumber | 3.1( getProductById) |
| Test Case Name | Test product functionality |
| Test Case description | corrrect 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 |  |

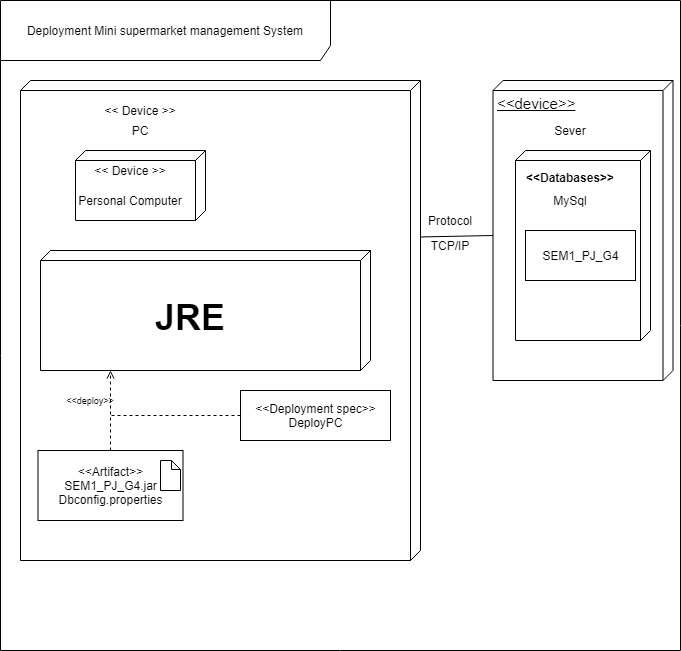
V. Assign work to each team member

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 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](https://www.oracle.com/technetwork/java/javase/downloads/index.html?fbclid=IwAR1o2QYFfNFCkz5U93FqihCoWuhQmru-CoEiE60Zf7j2Yh6xORu_HaBUjOo)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:



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

1. 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

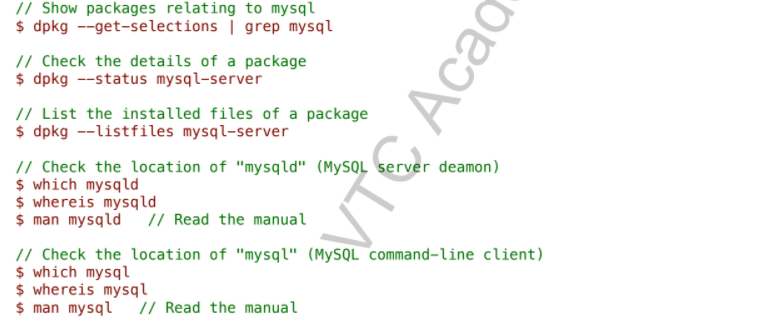
1. Installing MySQL on Linux/UNIX

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

$ sudo apt-get update

$ sudo apt-get install mysql-server

+ Verify the MySQL Installation

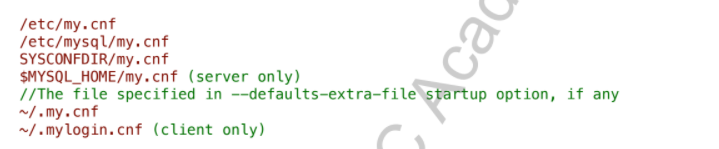


+ Configure MySQL Server

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>)



The installation default /etc/mysql/my.cnf includes directories /etc/mysql/

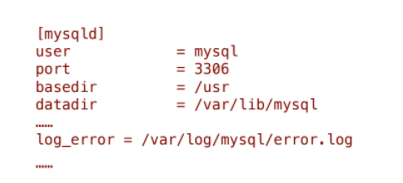
conf.d/ and /etc/mysql/mysql.conf.d/. The /etc/mysql/conf.d/mysql.cnf is

empty. Hence, the main configuaration 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:



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

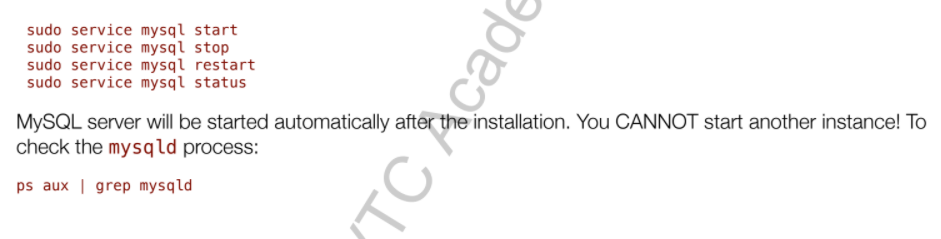
The server runs on the default port number of 3306.

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

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

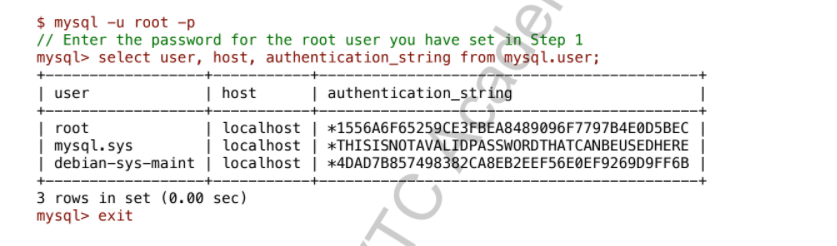
+ Start/Shutdown MySQL Server (mysqld)

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:



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

To start a MySQL client:



* Install App:

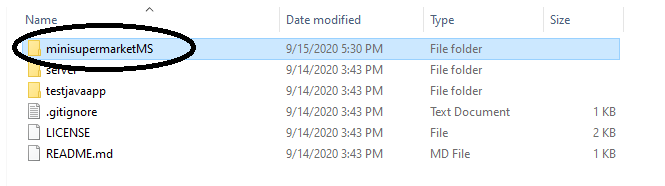
+ Follow link: github.com/ngonngay/Project

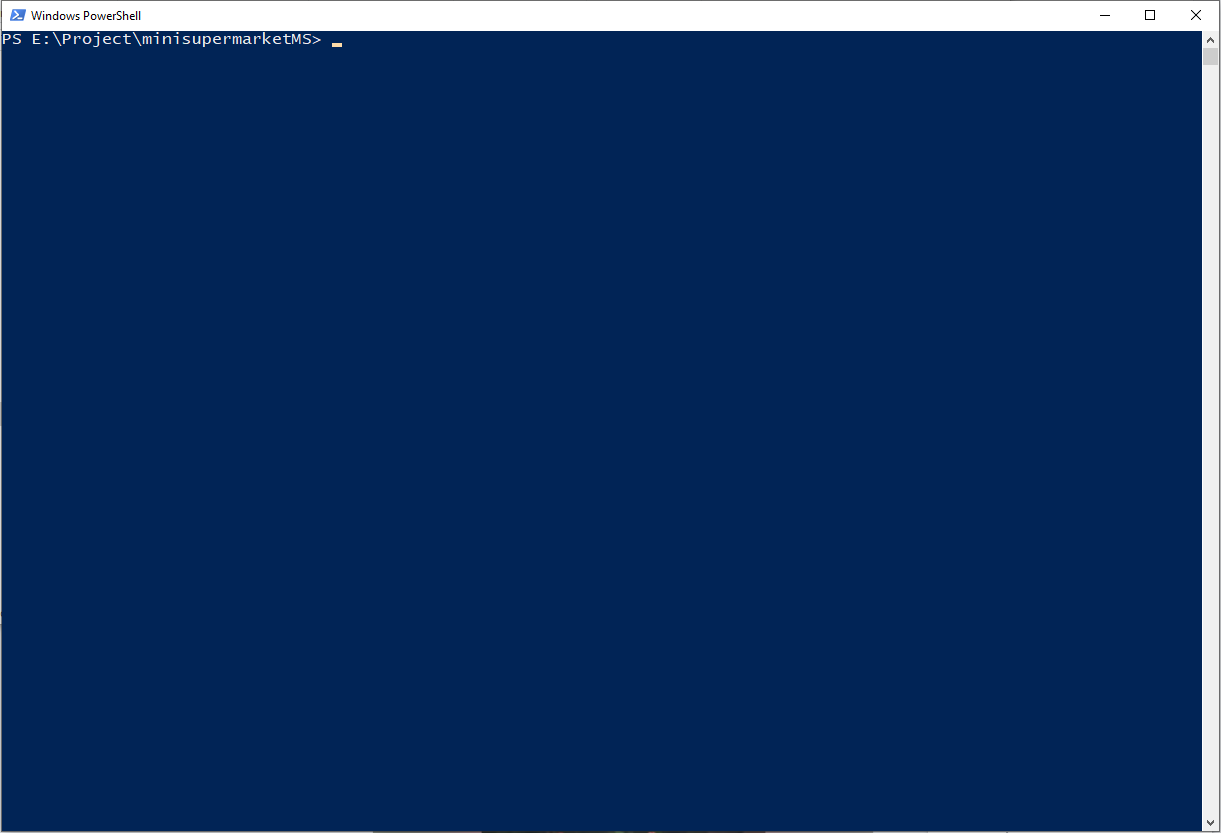
+ Download file ZIP

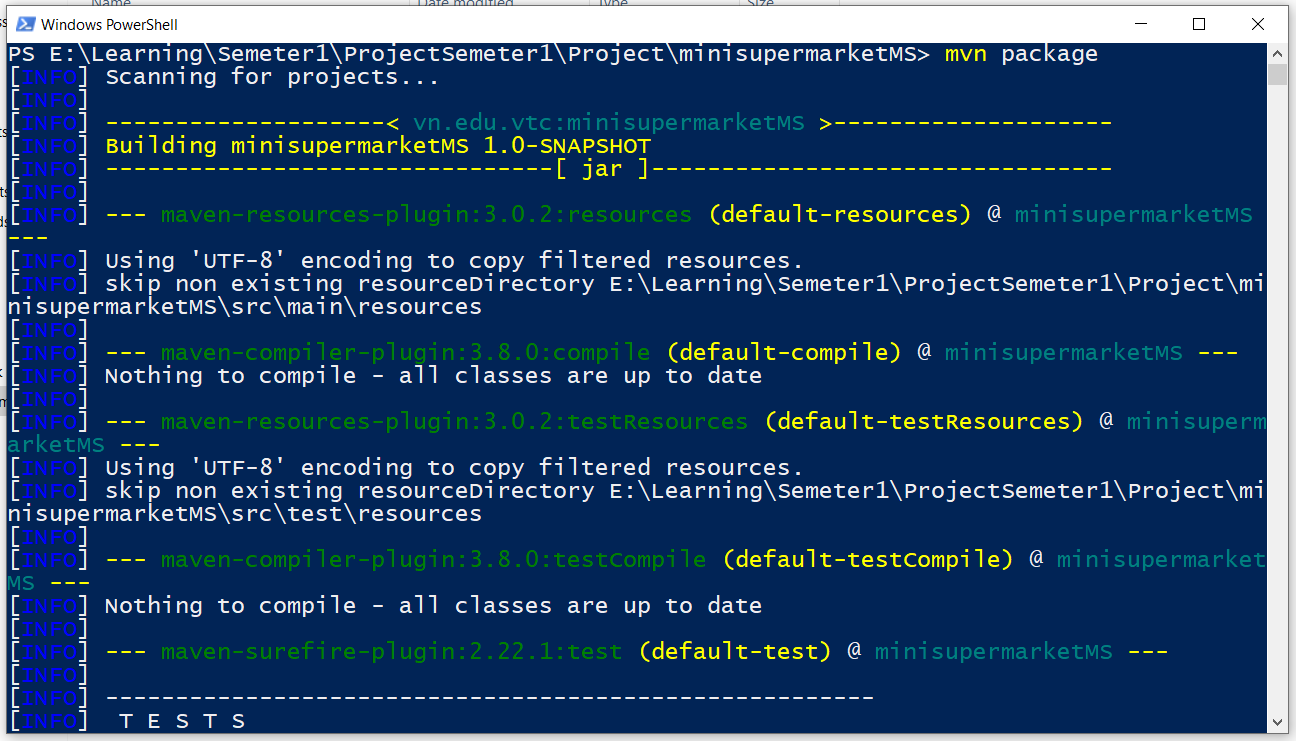
+ Unzip file

+ Biên dịch file:

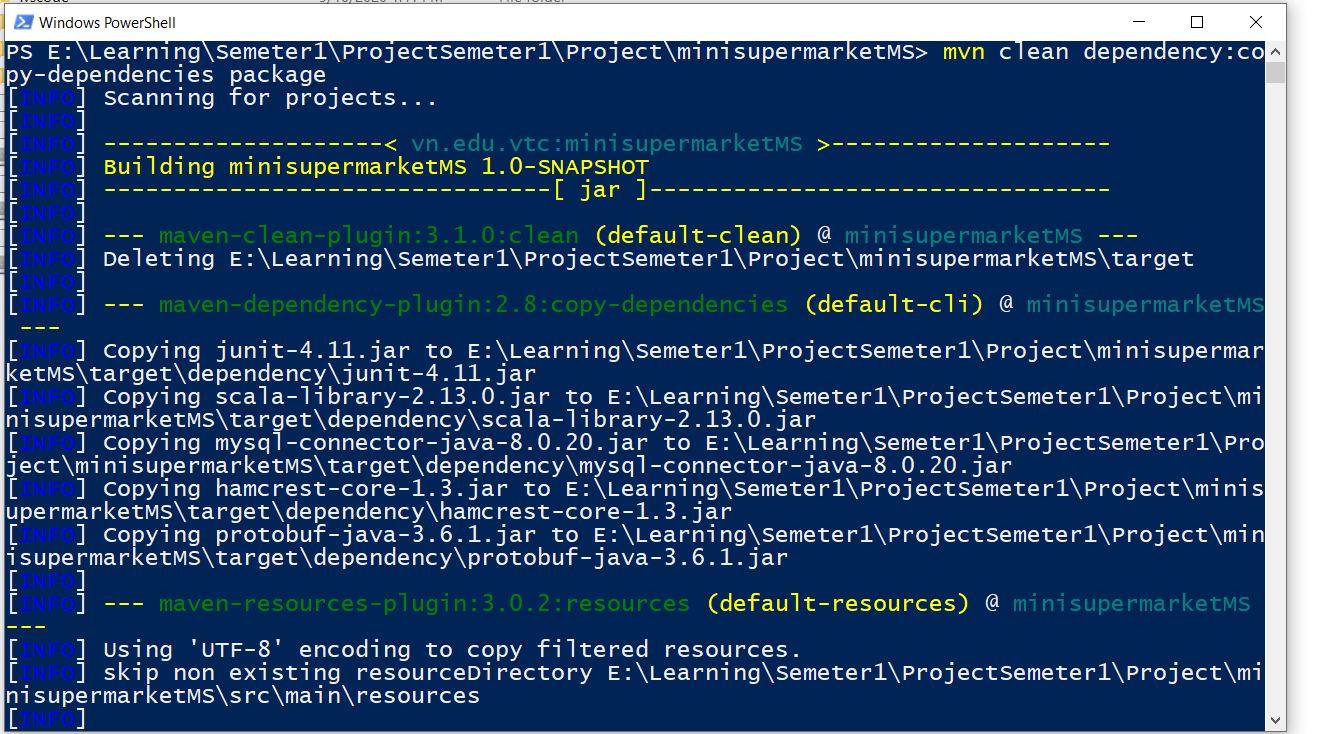
* Open folder “minisupermarketMS”



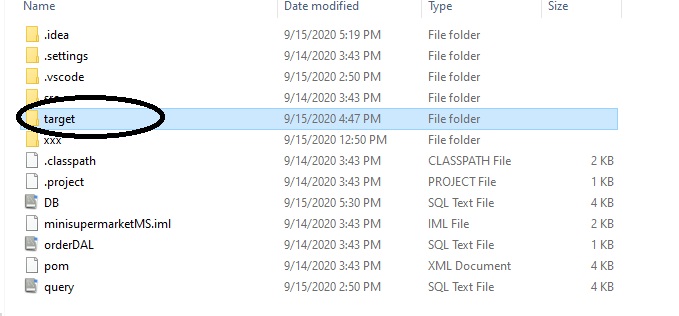
* Shift + right mouse to open PowerShell
* Type commands “mvn package”



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



* After open folder “target”



* Double-click to app icon