

**PROJECT REPORT**

# Sales System

|  |  |
| --- | --- |
| **Semester:** | Programming Fundamentals |
| **Class:** | PF13 |
| **Group:** | Group Name |
| **Instructor** | Instructor\_Name |
| **Team members:** | Trần Văn Tiến  Vũ Văn Phúc  Mạc Hồng Phước |

# Index

Project Name 1

Index 2

I. Project introduction 3

II. Analyze System Requirements 3

III. Design Details 8

IV. Test 14

V. Task Assign (to each team member) 15

VI. Installation Instructions 16

Appendix 17

Document format 18

# Project introduction

Briefly describe the operation of the system to which the project will be applied

* Store data about products, invoices. Support flexible calculation, check quantity, save time for users
* Fast data update

1. Proposed System

...

1. The scope of the project to be applied

* Shops selling milk tea, fast food

1. System Name

* Sales System

1. Deployment Environment

* PC, Laptop
* Windows 10, Linux, MacOS

1. Development Tools

* Visual Studio Code
* MySql Sever 8.0: Data saving
* Draw.io: Draw system design diagram
* Microsoft Word: Write a report

1. Customer Requirements

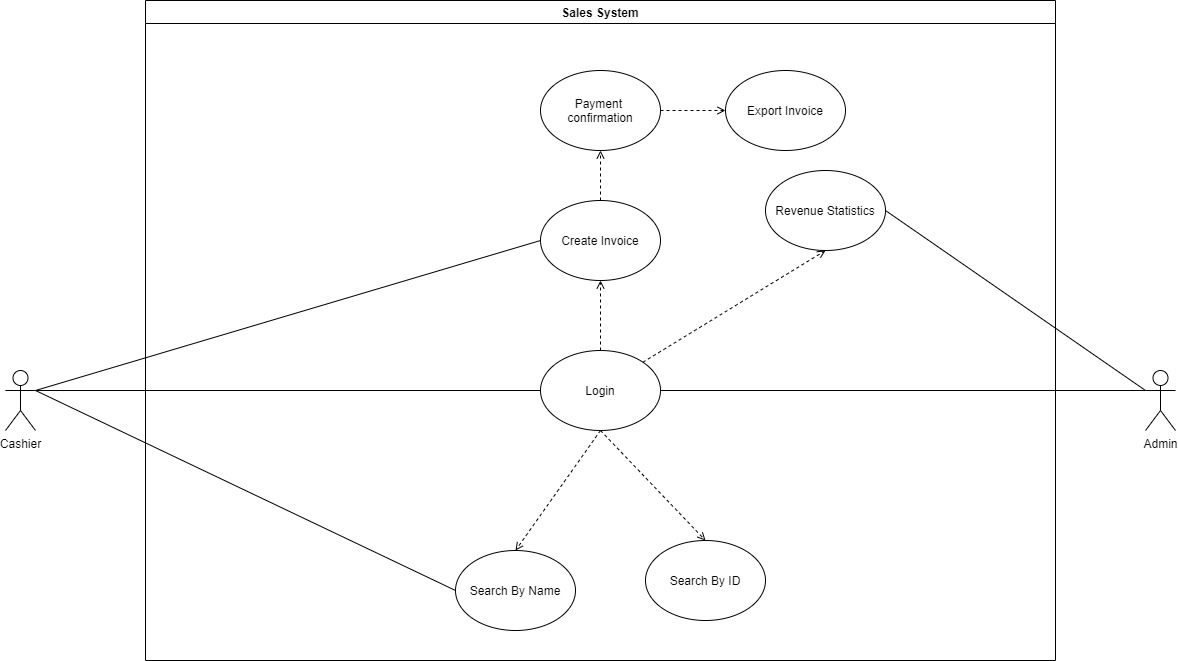
(System features)

* Manage product catalog
* Create invoice
* Export and print invoices

# Analyze System Requirements

< Specify the requirements of the system required to meet customer requirements. This is the content of the discussion presented in more detail >

1. Use Case



|  |  |
| --- | --- |
| Name | Login |
| ID | UC-01 |
| Description | User login to the system |
| Actor | Seller |
| Organizational  Benefits | Log in to the system |
| Frequency of Use | Every time you use the system. |
| Triggers | User wants to login to the application. |
| Preconditions | - User accounts already created |
| Postconditions | - User successfully login to the application  - The system records successful login activity |
| Main Courses | 1. User enter account name, password 2. Check account information 3. Show main menu. |
| Alternate Courses | AC2.1 User enters wrong account password   1. The system notifies the wrong account or password 2. Show the login screen again 3. Re-enter request   AC2.2 Username password   1. Show successful login message 2. Show main menu |
| Exception | EX1.1 Invalid user name/password   1. Display re-entry message 2. Re-enter   EX2.1: Lost database connection   1. Show server connection lost message 2. Return to the login screen |

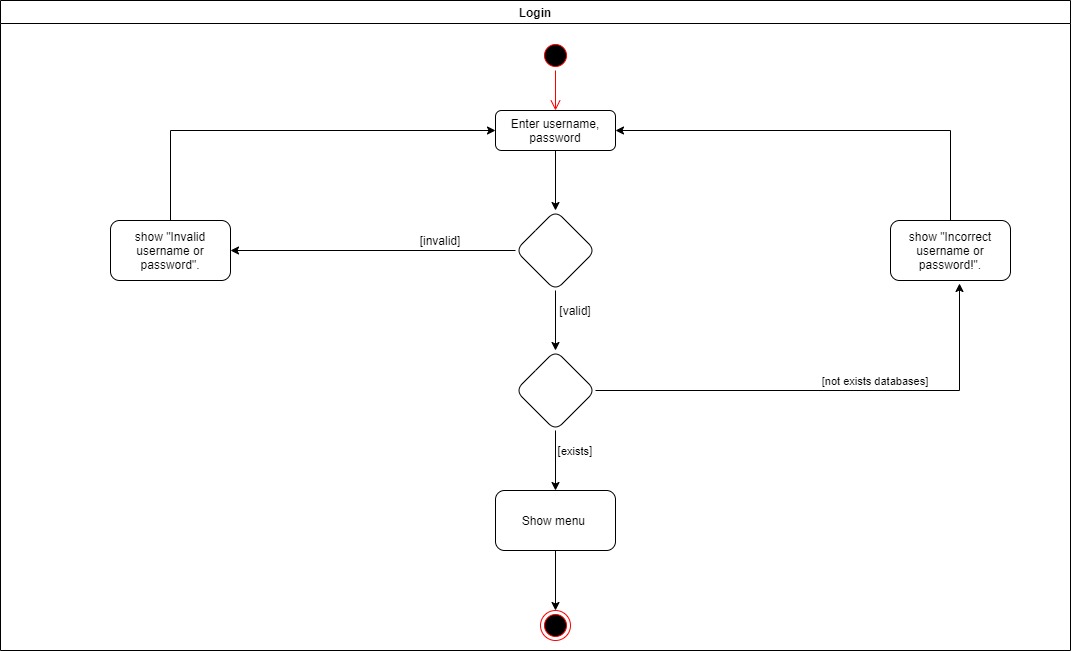
|  |  |
| --- | --- |
| Name | SearchByID |
| ID | UC-02 |
| Description | Find products by ID |
| Actor | Cashier |
| Organizational  Benefits | Shorten product search time |
| Triggers | Search for a product when the product id is known |
| Preconditions | - Logged in to the system |
| Postconditions | - Return product if found  - "Not Found" message if the product is not found |
| Main Courses | 1. Vendors log in to the system  2. The seller chooses the search section  3. Seller chooses "Search By ID"  4. Seller enters search product id  5. Show product details |
| Alternate Courses | AC4.1 No products found   1. Display the message "No products found" 2. Back to feature “Search By ID” |
| Exception | EX4.1 Lost database connection   1. Show “Lost database connection” 2. Back to feature “Search By ID”. |

|  |  |
| --- | --- |
| Name | SearchByName |
| ID |  |
| Description | Product search by product name |
| Actor | Cashier |
| Organizational  Benefits | Search list of products by name. |
| Triggers | Product search when know name |
| Preconditions | - Logged into the system |
| Postconditions | - Show list of products |
| Main Courses | 1. Vendors log in to the system  2. The seller chooses the search section  3. Seller chooses "Search By Name"  4. Seller enters search product name  5. Show product list. |
| Alternate Courses | AC4.1 No products found   1. Display the message "No products found" 2. Back to feature “Search By Name” |
| Exception | EX4.1 Lost database connection   1. Show “Lost database connection” 2. Back to feature “Search By Name”. |

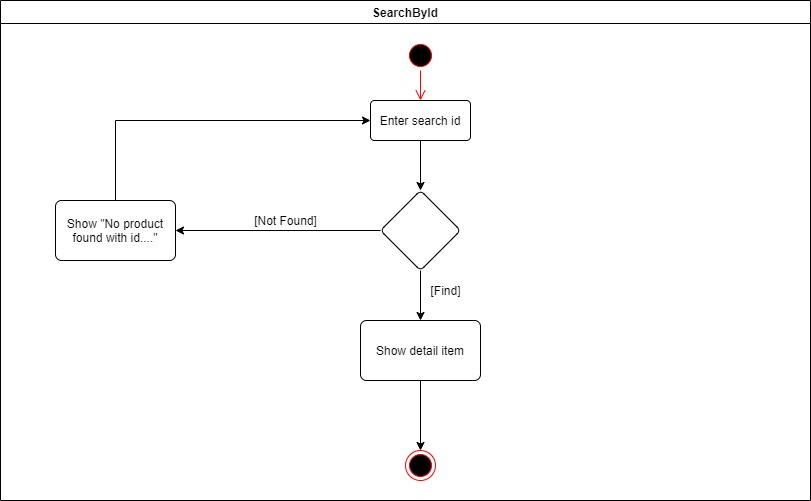
|  |  |
| --- | --- |
| Name | Create Invoice |
| ID | UC-02 |
| Description | Create detailed invoices for products ordered from customers. |
| Actor | Cashier |
| Organizational  Benefits | Save time for the payment. |
| Frequency of Use | Every time serving customers. |
| Triggers | Customer pays for the order. |
| Preconditions | - Logged into the system. |
| Postconditions | - Login to the system successfully.  - Invoice recording system. |
| Main Courses | 1. The seller logs into the system.  2. User selects invoice creation feature.  3. Successful invoice creation on the system to save information.  4. Show invoice. |
| Alternate Courses | AC3.1 Invoice creation failed due to lack of goods   1. Display out of stock error message 2. Does not allow the invoice to be generated and requires re-entry. |
| Exception |  |

1. Activity Diagram:

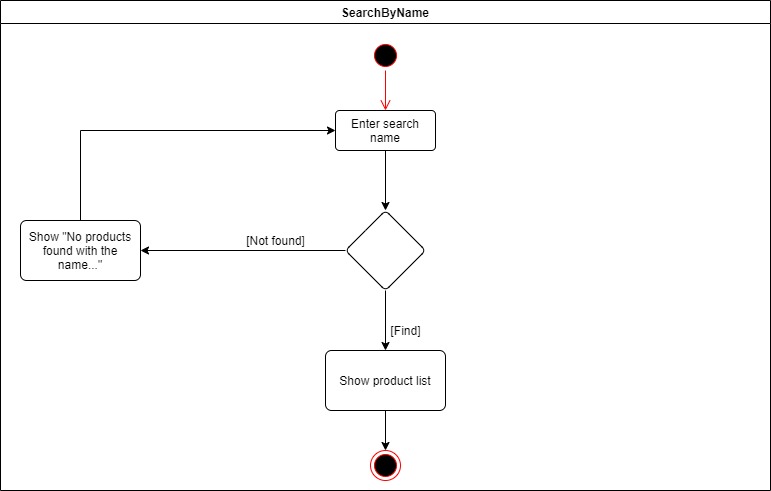
* Login



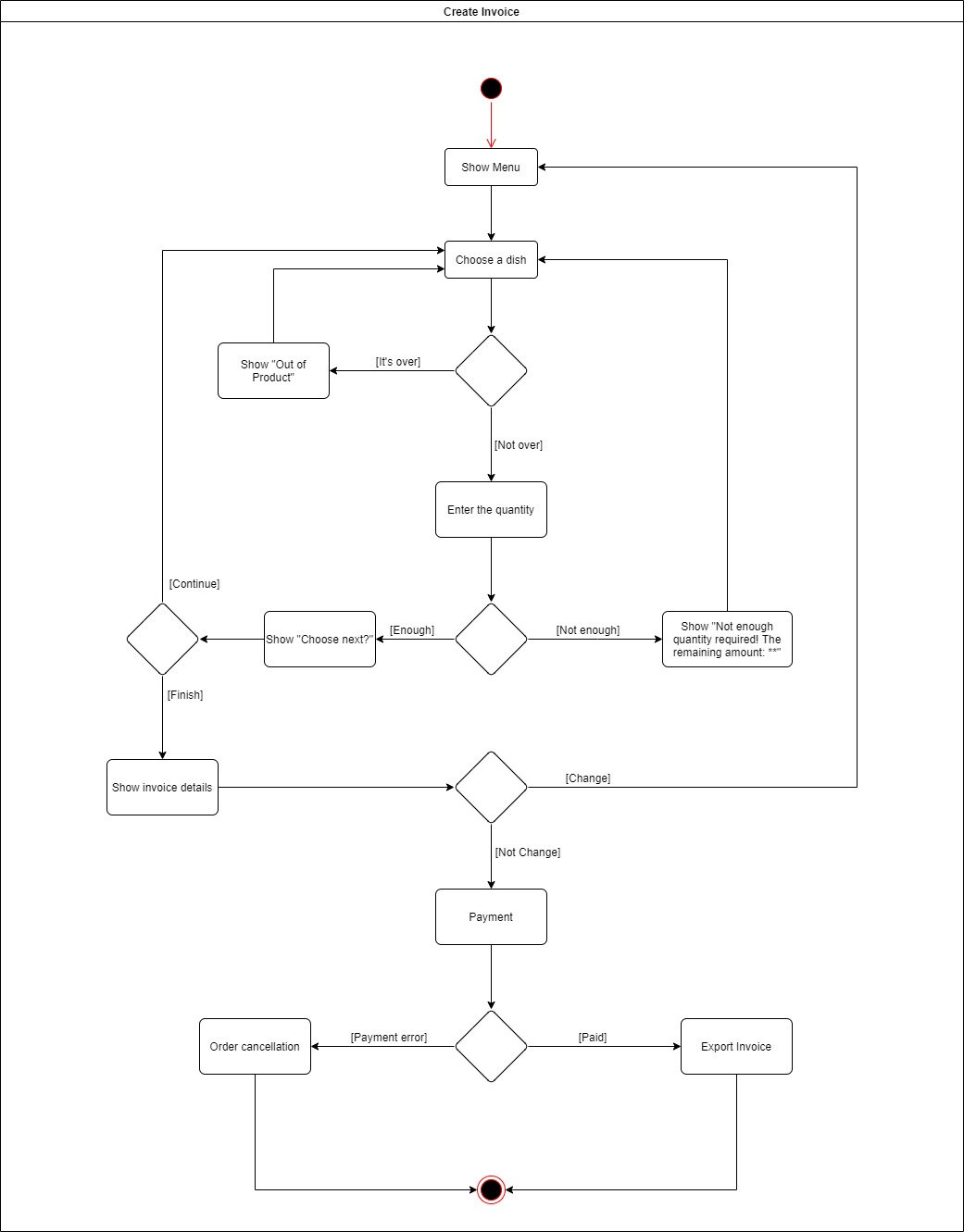
* Search By ID



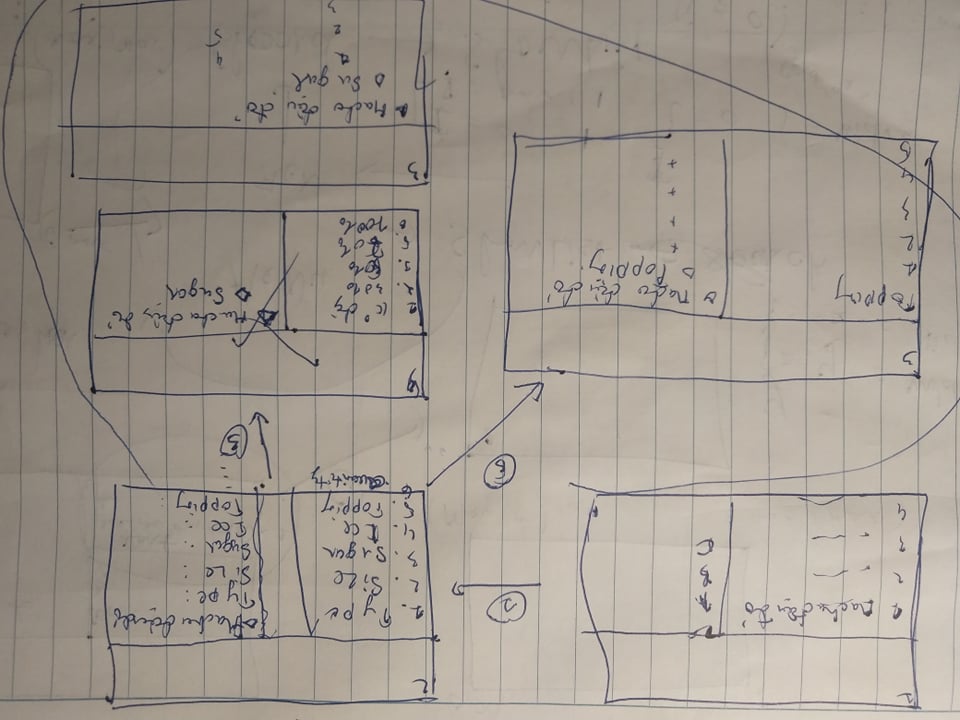
* Search By Name

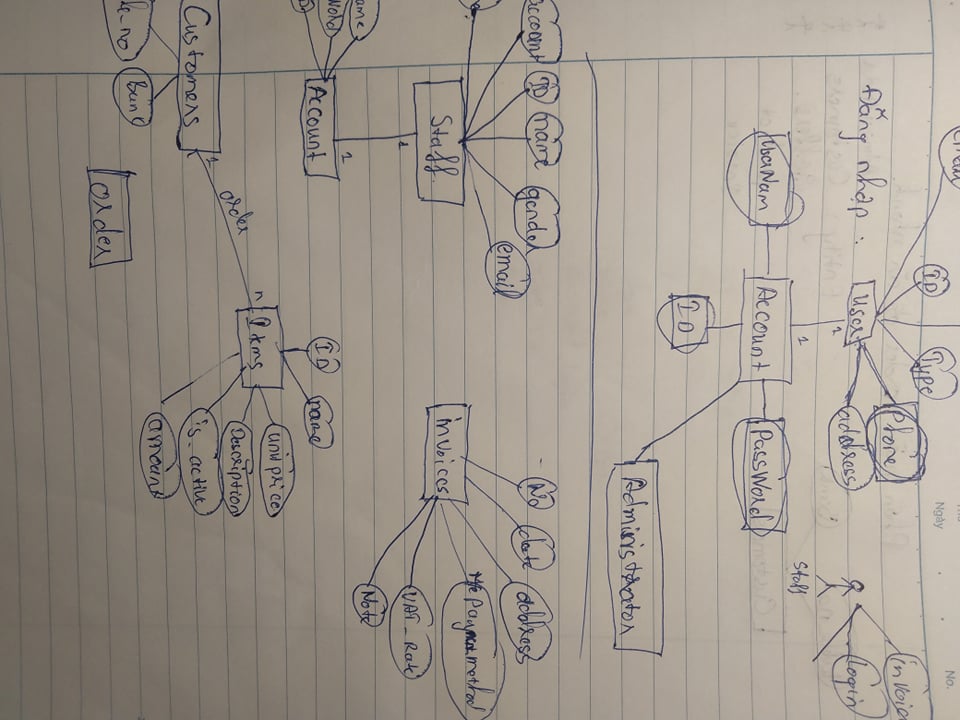
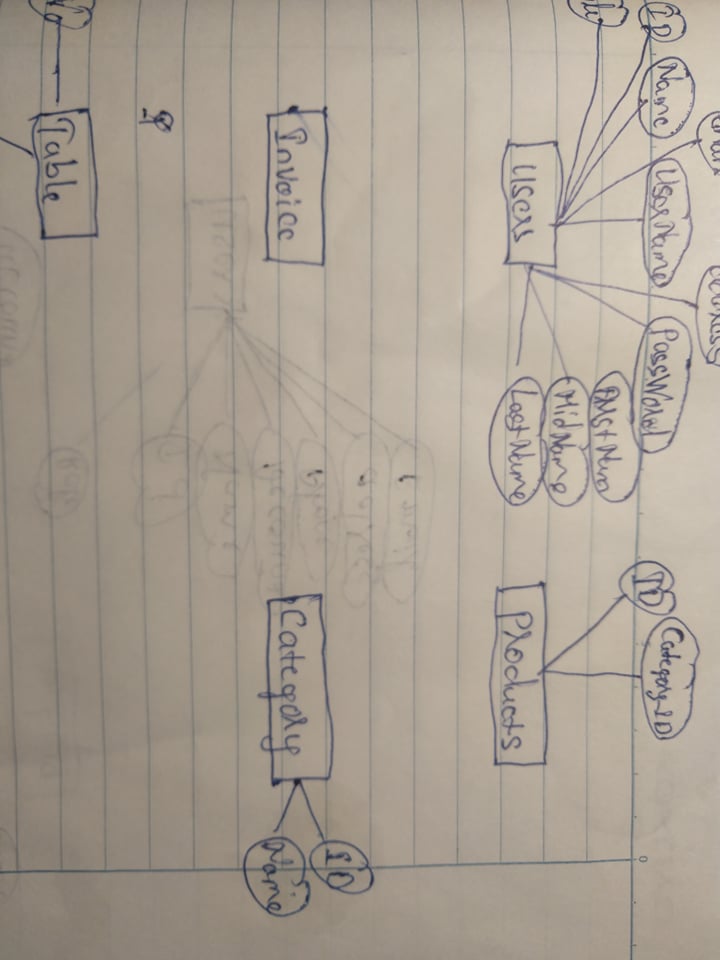


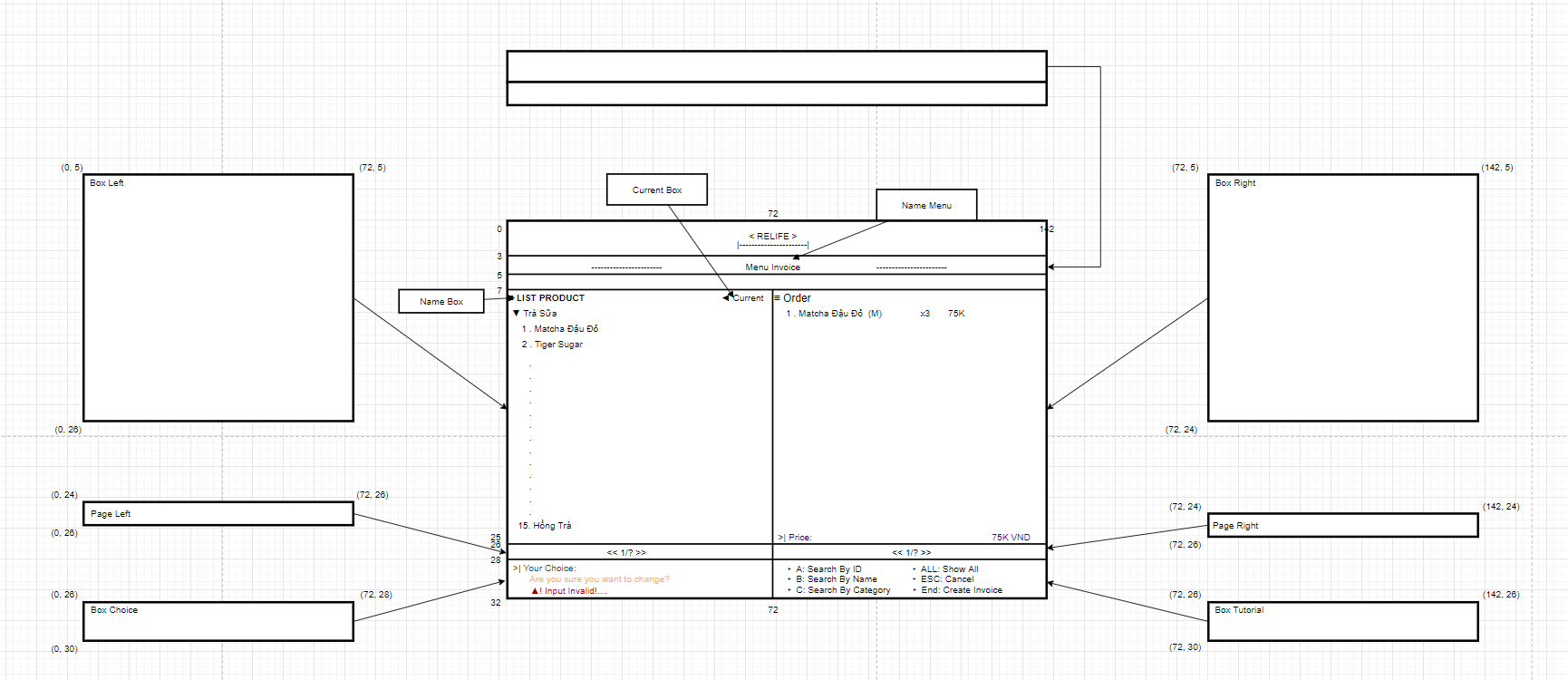
* Create Invoice

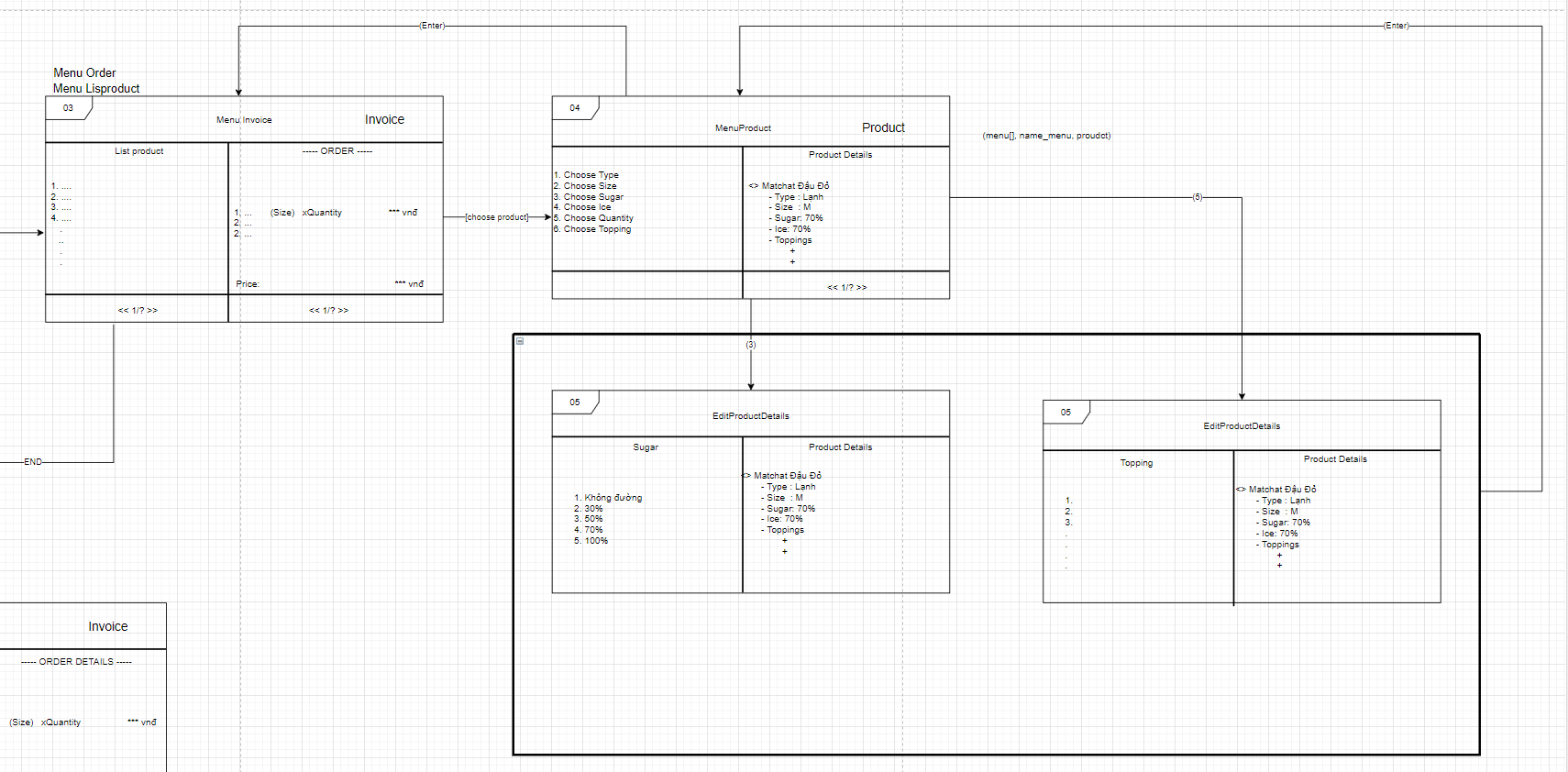


# Design Details

1. UI Design

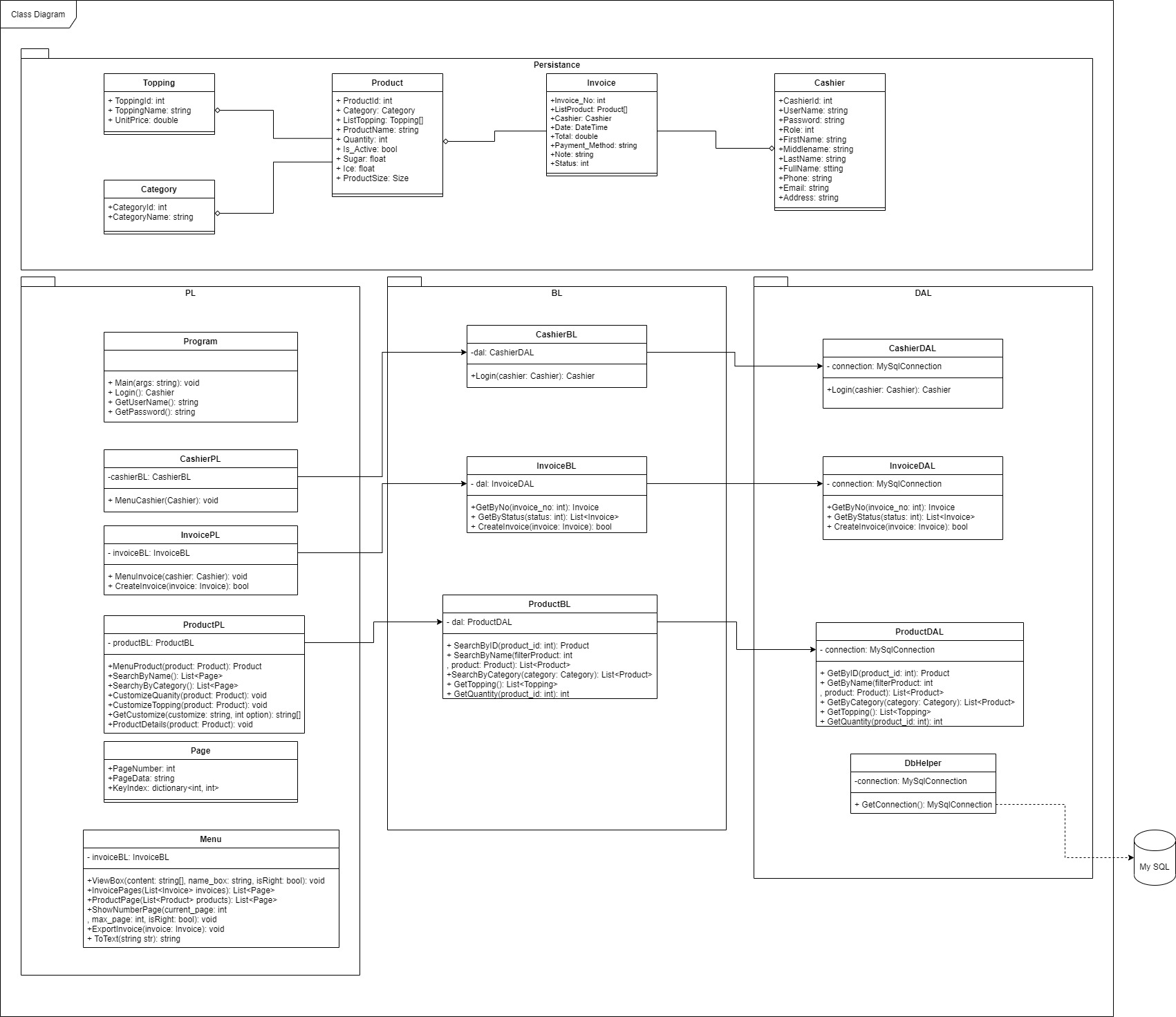
 





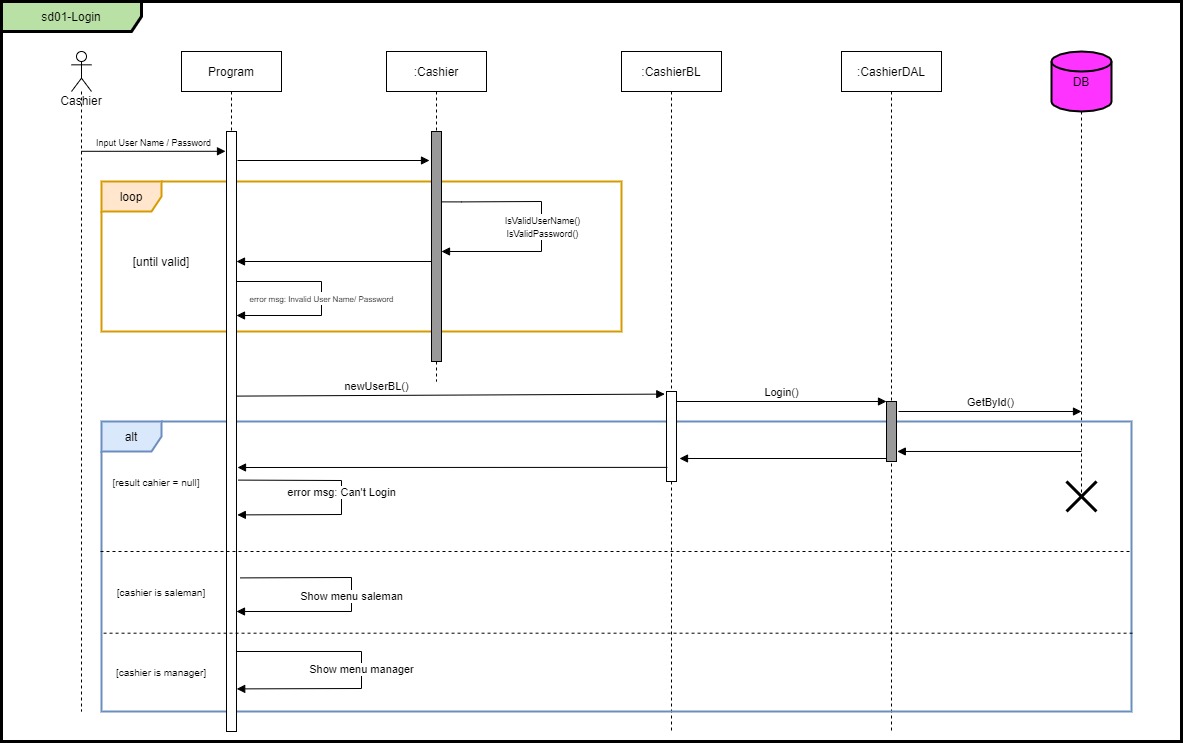
2. Code Design (Class Diagram)

(Class Diagram):

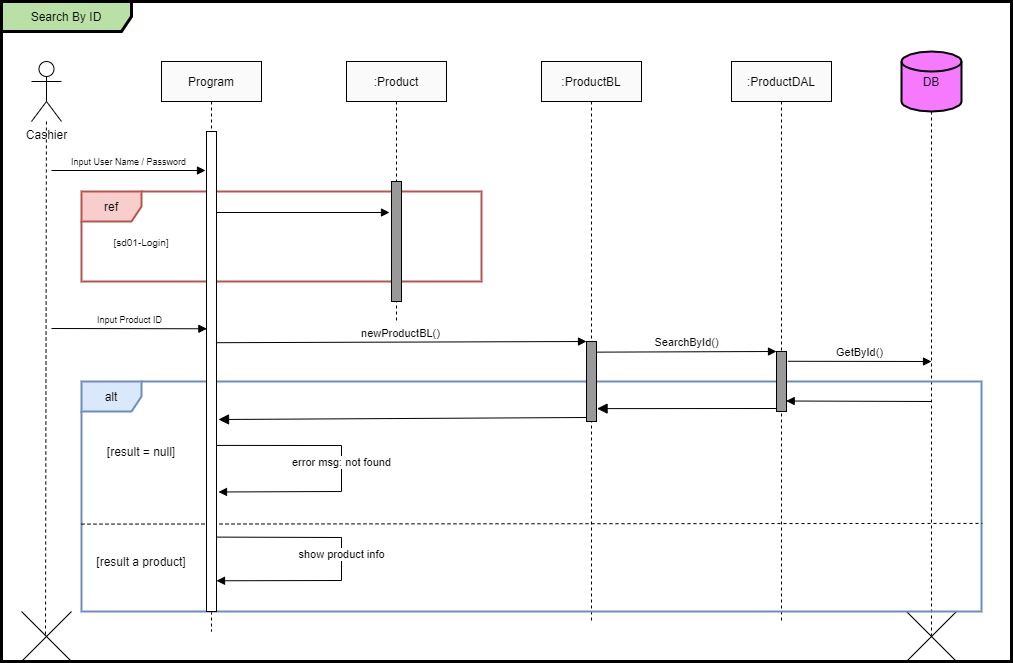


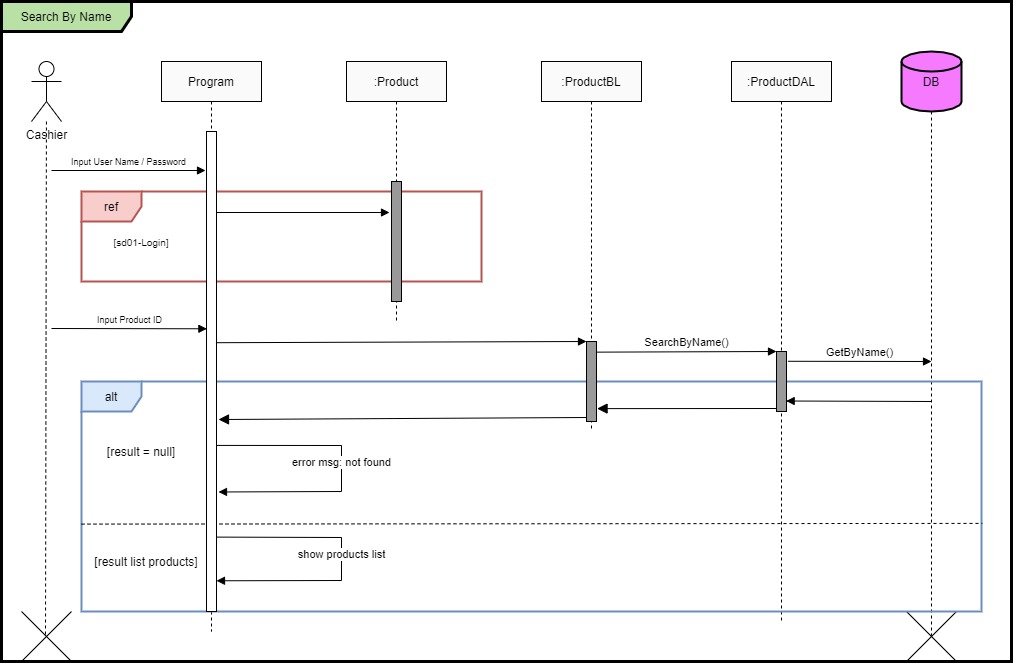
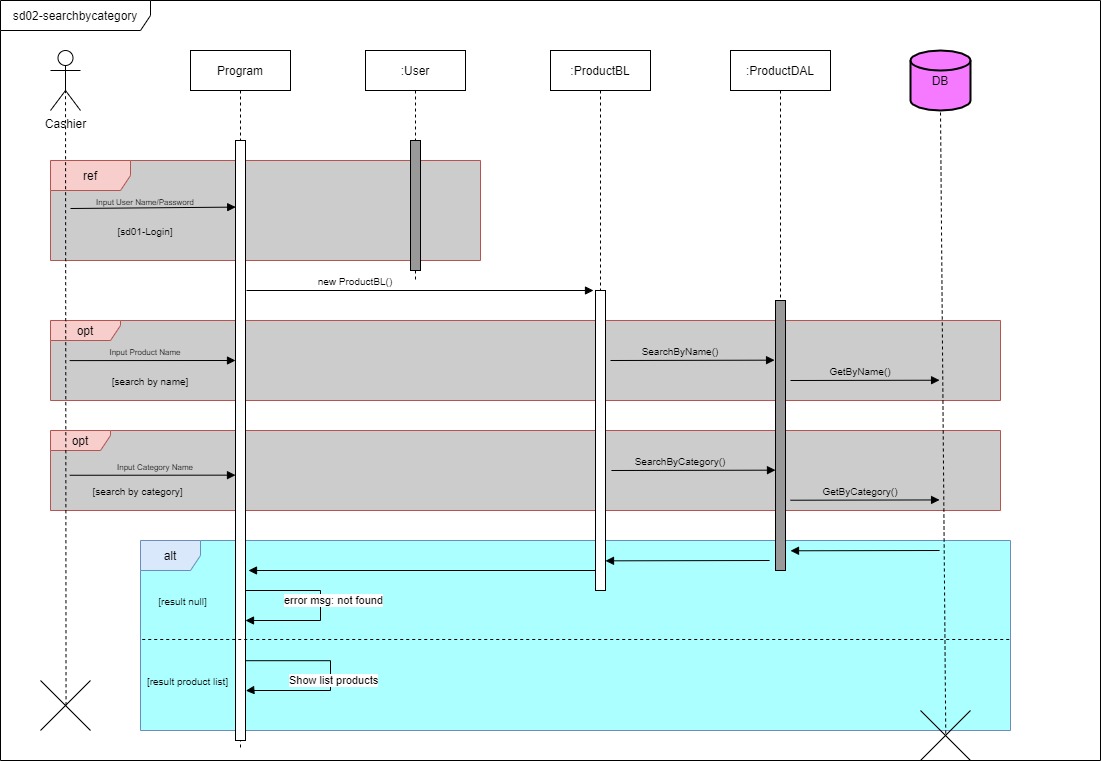
1. Sequence Diagram

* Login

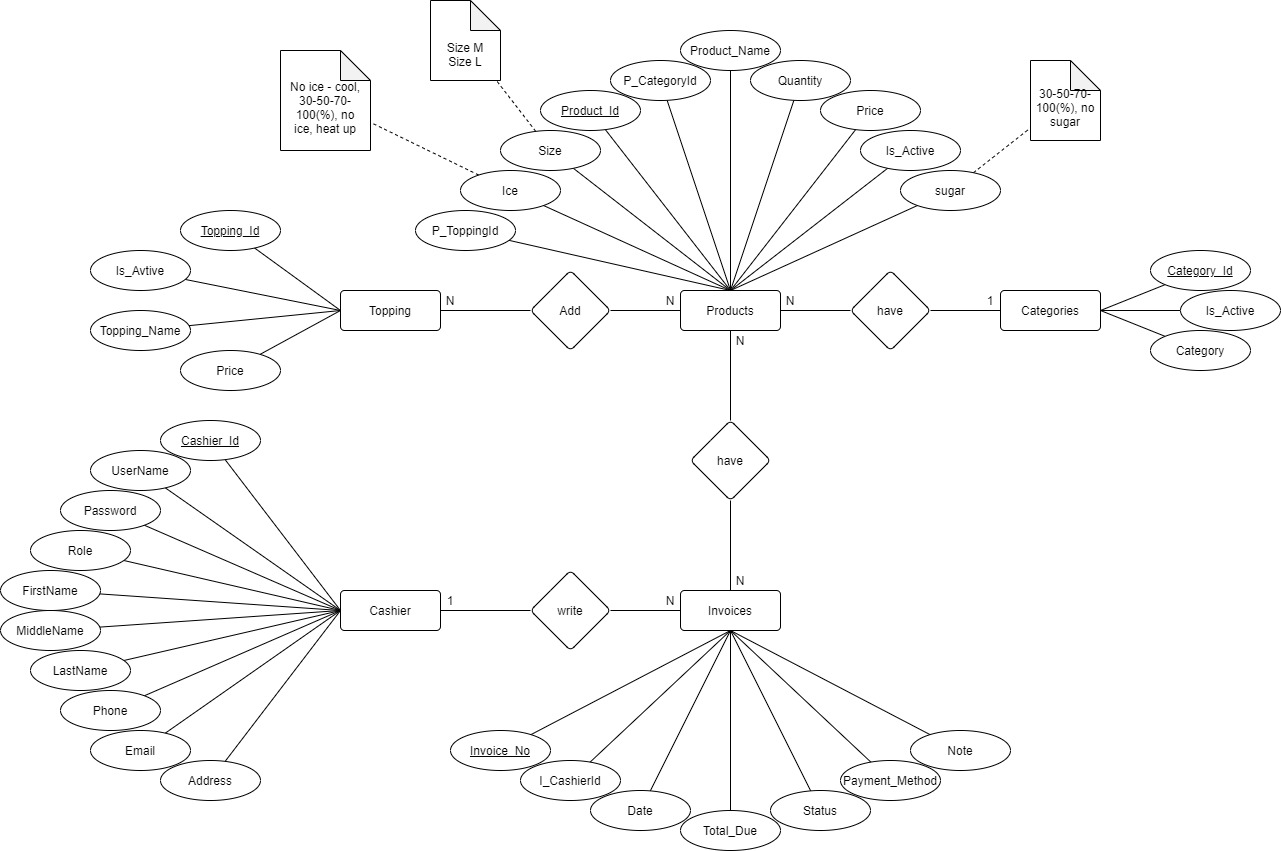


* Search By ID

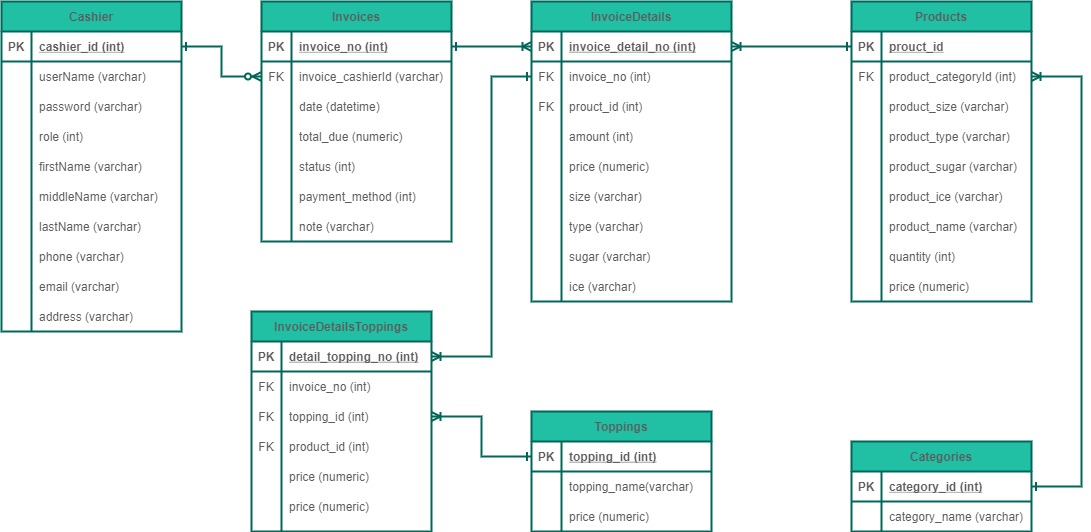


* Search By Name
* Search By Category

1. Database Design
   1. Entity Relationship Diagram



* 1. Database Design Details



|  |  |  |  |
| --- | --- | --- | --- |
| Cashier | | | |
| Column Name | Data Type | Constraints | Description |
| Cashier\_id | int | Primary Key, Autoincrement |  |
| UserName | varchar(50) | NOT NULL |  |
| Password | varchar(50) | NOT NULL |  |
| FirstName | varchar(50) | NOT NULL |  |
| MiddleName | varchar(50) | NOT NULL |  |
| LastName | varchar(50) | NOT NULL |  |
| Email | varchar(100) | NOT NULL | User Email |
| Phone | varchar(11) |  |  |
| Role | int | NOT NULL |  |
| Address | varchar(50) | NOT NULL | User Address |

|  |  |  |  |
| --- | --- | --- | --- |
| Invoice | | | |
| Column Name | Data Type | Constraints | Description |
| Invoice\_no | int | Primary Key, Autoincrement |  |
| Invoice\_cashier | varchar(50) | NOT NULL, Foreign key |  |
| Total\_due | numeric | NOT NULL |  |
| Status | int | DEFAULT 0 | Status of the invoice |
| Payment\_method | int | DEFAULT 0 |  |
| Note | Varchar(150) |  |  |
| Date | Date | Not null | Get the current system time |

|  |  |  |  |
| --- | --- | --- | --- |
| Product | | | |
| Column Name | Data Type | Constraints | Description |
| Product\_id | int | Primary Key, Autoincrement |  |
| Product\_category | varchar(50) | NOT NULL, foreign key |  |
| Product\_size | varchar(10) | NOT NULL | Size M, Size L |
| Product\_type | varchar(10) | NOT NULL | Hot, Cold |
| Product\_sugar | varchar(10) | NOT NULL | 30%, 50%, 70%, 100%, 0% |
| Product\_ice | varchar(10) | NOT NULL | 30%, 50%, 70%, 100%, 0%, Heat up, No ice - cool |
| Product\_name | varchar(50) | NOT NULL |  |
| Price | numeric | NOT NULL |  |
| Quantity | int | NOT NULL |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Category | | | |
| Column Name | Data Type | Constraints | Description |
| Category\_id | int | Primary Key, Autoincrement |  |
| Catgegory\_name | varchar(50) | Not null |  |

|  |  |  |  |
| --- | --- | --- | --- |
| InvoiceDetailTopping | | | |
| Column Name | Data Type | Constraints | Description |
| Detail\_topping\_no | int | Primary Key, Autoincrement |  |
| Invoice\_no | int | NOT NULL, foreign key |  |
| Topping\_id | int | NOT NULL, foreign key |  |
| Product\_id | int | NOT NULL, foreign key |  |
| Price | numeric | NOT NULL |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Topping | | | |
| Column Name | Data Type | Constraints | Description |
| Topping\_id | int | Primary Key |  |
| Topping\_name | varchar(50) | NOT NULL |  |
| Price | numeric | NOT NULL |  |

|  |  |  |  |
| --- | --- | --- | --- |
| InvoiceDetail | | | |
| Column Name | Data Type | Constraints | Description |
| Invoice\_detail\_no | int | Primary Key, Autoincrement |  |
| Invoice\_no | int | NOT NULL |  |
| Amount | int | NOT NULL |  |
| Product\_id | int | NOT NULL |  |
| Price | numeric | NOT NULL |  |
| Size | varchar(10) | NOT NULL |  |
| Type | varchar(10) | NOT NULL |  |
| Ice | varchar(10) | NOT NULL |  |
| Sugar | varchar(10) | NOT NULL |  |

# Test

|  |  |
| --- | --- |
| Test Case Number | 1.1 |
| Test Case Name | LoginTest |
| Test Case Description | This test case check user's password account |
| Preconditions | Already have an account |
| Test Case Input | Enter:  - User Name: Administrator   * Password: AdiminPF13 |
| Test Case Expected Output | Role = 1 |
| Test Case Steps | Step 1: Enter username and password  Step 2: Check database  Step 3: Return cashier with corresponding role |
|  |  |

|  |  |
| --- | --- |
| Test Case Number | 1.2 |
| Test Case Name | LoginTest |
| Test Case Description | This test case check user's password account |
| Preconditions | Already have an account |
| Test Case Input | Enter:  - User Name: Tientv   * Password: TienPF13 |
| Test Case Expected Output | Role = 2 |
| Test Case Steps | Step 1: Enter username and password  Step 2: Check database  Step 3: Return cashier with corresponding role |
|  |  |

|  |  |
| --- | --- |
| Test Case Number | 1.3 |
| Test Case Name | LoginTest |
| Test Case Description | This test case check user's password account |
| Preconditions | Already have an account |
| Test Case Input | Enter:  - User Name: Phucvv   * Password: abcde |
| Test Case Expected Output | Role = 0 |
| Test Case Steps | Step 1: Enter username and password  Step 2: Check database  Step 3: Return cashier with corresponding role |
|  |  |

|  |  |
| --- | --- |
| Test Case Number | 1.4 |
| Test Case Name | LoginTest |
| Test Case Description | This test case check user's password account |
| Preconditions | Already have an account |
| Test Case Input | Enter:  - User Name: Phuocabc   * Password: PhuocPF13 |
| Test Case Expected Output | Role = 0 |
| Test Case Steps | Step 1: Enter username and password  Step 2: Check database  Step 3: Return cashier with corresponding role |
|  |  |

|  |  |
| --- | --- |
| Test Case Number | 1.5 |
| Test Case Name | LoginTest |
| Test Case Description | This test case check user's password account |
| Preconditions | Already have an account |
| Test Case Input | Enter:  - User Name: Tienabc   * Password: Tiemabc |
| Test Case Expected Output | Role = 0 |
| Test Case Steps | Step 1: Enter username and password  Step 2: Check database  Step 3: Return cashier with corresponding role |
|  |  |

|  |  |
| --- | --- |
| Test Case Number | 2.1 |
| Test Case Name | GetByID1 |
| Test Case Description | This test case search cashier by ID |
| Preconditions | Login with account admin |
| Test Case Input | Enter id = 1 |
| Test Case Expected Output | Information of a respective employee |
| Test Case Steps | Step 1: Enter employee ID  Step 2: Check database  Step 3: Information of a respective employee |
|  |  |

|  |  |
| --- | --- |
| Test Case Number | 2.2 |
| Test Case Name | GetByID1 |
| Test Case Description | This test case search cashier by ID |
| Preconditions | Login with account admin |
| Test Case Input | Enter id 2 |
| Test Case Expected Output | Information of a respective employee |
| Test Case Steps | Step 1: Enter employee ID  Step 2: Check database  Step 3: Information of a respective employee |
|  |  |

|  |  |
| --- | --- |
| Test Case Number | 2.3 |
| Test Case Name | GetByID2 |
| Test Case Description | This test case search cashier by ID |
| Preconditions | Login with account admin |
| Test Case Input | Enter id -1 |
| Test Case Expected Output | Null value |
| Test Case Steps | Step 1: Enter employee ID  Step 2: Check database  Step 3: Information of a respective employee |
|  |  |

|  |  |
| --- | --- |
| Test Case Number | 2.4 |
| Test Case Name | GetByID2 |
| Test Case Description | This test case search cashier by ID |
| Preconditions | Login with account admin |
| Test Case Input | Enter id = 0 |
| Test Case Expected Output | Null value |
| Test Case Steps | Step 1: Enter employee ID  Step 2: Check database  Step 3: Information of a respective employee |
|  |  |

|  |  |
| --- | --- |
| Test Case Number | 2.5 |
| Test Case Name | GetByID2 |
| Test Case Description | This test case search cashier by ID |
| Preconditions | Login with account admin |
| Test Case Input | Enter id = 999999999 |
| Test Case Expected Output | Null value |
| Test Case Steps | Step 1: Enter employee ID  Step 2: Check database  Step 3: Information of a respective employee |
|  |  |

|  |  |
| --- | --- |
| Test Case Number | 3.1 |
| Test Case Name | GetByIDTest1 |
| Test Case Description | This test case search Product by ID |
| Preconditions | Logged in user |
| Test Case Input | Enter id Product = 1 |
| Test Case Expected Output | Return a product |
| Test Case Steps | Step 1: Enter product ID  Step 2: Check database  Step 3: Return a product |
|  |  |

|  |  |
| --- | --- |
| Test Case Number | 3.2 |
| Test Case Name | GetByIDTest1 |
| Test Case Description | This test case search Product by ID |
| Preconditions | Logged in user |
| Test Case Input | Enter id Product = 50 |
| Test Case Expected Output | Return a product |
| Test Case Steps | Step 1: Enter product ID  Step 2: Check database  Step 3: Return a product |
|  |  |

|  |  |
| --- | --- |
| Test Case Number | 3.3 |
| Test Case Name | GetByIDTest1 |
| Test Case Description | This test case search Product by ID |
| Preconditions | Logged in user |
| Test Case Input | Enter id Product = 15 |
| Test Case Expected Output | Return a product |
| Test Case Steps | Step 1: Enter product ID  Step 2: Check database  Step 3: Return a product |
|  |  |

|  |  |
| --- | --- |
| Test Case Number | 3.4 |
| Test Case Name | GetByIDTest2 |
| Test Case Description | This test case search Product by ID |
| Preconditions | Logged in user |
| Test Case Input | Enter id Product = -1 |
| Test Case Expected Output | Null value |
| Test Case Steps | Step 1: Enter product ID  Step 2: Check database  Step 3: Return a product |
|  |  |

|  |  |
| --- | --- |
| Test Case Number | 3.5 |
| Test Case Name | GetByIDTest2 |
| Test Case Description | This test case search Product by ID |
| Preconditions | Logged in user |
| Test Case Input | Enter id Product = 0 |
| Test Case Expected Output | Null value |
| Test Case Steps | Step 1: Enter product ID  Step 2: Check database  Step 3: Return a product |
|  |  |

|  |  |
| --- | --- |
| Test Case Number | 3.6 |
| Test Case Name | GetByIDTest2 |
| Test Case Description | This test case search Product by ID |
| Preconditions | Logged in user |
| Test Case Input | Enter id Product = 100000000 |
| Test Case Expected Output | Null value |
| Test Case Steps | Step 1: Enter product ID  Step 2: Check database  Step 3: Return a product |
|  |  |

|  |  |
| --- | --- |
| Test Case Number | 4.1 |
| Test Case Name | GetProductTest1 |
| Test Case Description | This test case search Product by name |
| Preconditions | Logged in user |
| Test Case Input | Enter product name = “Okinawa” |
| Test Case Expected Output | Return list Product |
| Test Case Steps | Step 1: Enter product ID  Step 2: Check database  Step 3: Return list Product |
|  |  |

|  |  |
| --- | --- |
| Test Case Number | 4.2 |
| Test Case Name | GetProductTest1 |
| Test Case Description | This test case search Product by name |
| Preconditions | Logged in user |
| Test Case Input | Enter product name = “nawa” |
| Test Case Expected Output | Return list Product |
| Test Case Steps | Step 1: Enter product ID  Step 2: Check database  Step 3: Return list Product |
|  |  |

|  |  |
| --- | --- |
| Test Case Number | 4.3 |
| Test Case Name | GetProductTest1 |
| Test Case Description | This test case search Product by name |
| Preconditions | Logged in user |
| Test Case Input | Enter product name = “kim” |
| Test Case Expected Output | Return list Product |
| Test Case Steps | Step 1: Enter product ID  Step 2: Check database  Step 3: Return list Product |
|  |  |

|  |  |
| --- | --- |
| Test Case Number | 4.4 |
| Test Case Name | GetProductTest1 |
| Test Case Description | This test case search Product by name |
| Preconditions | Logged in user |
| Test Case Input | Enter product name = “Trà Sữa” |
| Test Case Expected Output | Return list Product |
| Test Case Steps | Step 1: Enter product ID  Step 2: Check database  Step 3: Return list Product |
|  |  |

|  |  |
| --- | --- |
| Test Case Number | 4.5 |
| Test Case Name | GetProductTest1 |
| Test Case Description | This test case search Product by name |
| Preconditions | Logged in user |
| Test Case Input | Enter product name = “Ô Long Trân Châu Baby Kem Cafe” |
| Test Case Expected Output | Return list Product |
| Test Case Steps | Step 1: Enter product ID  Step 2: Check database  Step 3: Return list Product |
|  |  |

|  |  |
| --- | --- |
| Test Case Number | 4.6 |
| Test Case Name | GetProductTest2 |
| Test Case Description | This test case search Product by name |
| Preconditions | Logged in user |
| Test Case Input | Enter product name = “1234” |
| Test Case Expected Output | Null value |
| Test Case Steps | Step 1: Enter product ID  Step 2: Check database  Step 3: Return list Product |
|  |  |

|  |  |
| --- | --- |
| Test Case Number | 4.7 |
| Test Case Name | GetProductTest1 |
| Test Case Description | This test case search Product by name |
| Preconditions | Logged in user |
| Test Case Input | Enter product name = “abcs” |
| Test Case Expected Output | Null value |
| Test Case Steps | Step 1: Enter product ID  Step 2: Check database  Step 3: Return list Product |
|  |  |

|  |  |
| --- | --- |
| Test Case Number | 5.1 |
| Test Case Name | GetByCategoryTest1 |
| Test Case Description | This test case search Product by category |
| Preconditions | Logged in user |
| Test Case Input | Enter category name = “Fresh” |
| Test Case Expected Output | Return list Product |
| Test Case Steps | Step 1: Enter product ID  Step 2: Check database  Step 3: Return list Product |
|  |  |

|  |  |
| --- | --- |
| Test Case Number | 5.2 |
| Test Case Name | GetByCategoryTest1 |
| Test Case Description | This test case search Product by category |
| Preconditions | Logged in user |
| Test Case Input | Enter category name = “fruit” |
| Test Case Expected Output | Return list Product |
| Test Case Steps | Step 1: Enter product ID  Step 2: Check database  Step 3: Return list Product |
|  |  |

|  |  |
| --- | --- |
| Test Case Number | 5.3 |
| Test Case Name | GetByCategoryTest1 |
| Test Case Description | This test case search Product by category |
| Preconditions | Logged in user |
| Test Case Input | Enter category name = “Trà Sữa” |
| Test Case Expected Output | Return list Product |
| Test Case Steps | Step 1: Enter product ID  Step 2: Check database  Step 3: Return list Product |
|  |  |

|  |  |
| --- | --- |
| Test Case Number | 5.4 |
| Test Case Name | GetByCategoryTest1 |
| Test Case Description | This test case search Product by category |
| Preconditions | Logged in user |
| Test Case Input | Enter category name = “Fresh Fruit Tea” |
| Test Case Expected Output | Return list Product |
| Test Case Steps | Step 1: Enter product ID  Step 2: Check database  Step 3: Return list Product |
|  |  |

|  |  |
| --- | --- |
| Test Case Number | 5.5 |
| Test Case Name | GetByCategoryTest1 |
| Test Case Description | This test case search Product by category |
| Preconditions | Logged in user |
| Test Case Input | Enter category name = “uit Tea” |
| Test Case Expected Output | Return list Product |
| Test Case Steps | Step 1: Enter product ID  Step 2: Check database  Step 3: Return list Product |
|  |  |

|  |  |
| --- | --- |
| Test Case Number | 5.6 |
| Test Case Name | GetByCategoryTest1 |
| Test Case Description | This test case search Product by category |
| Preconditions | Logged in user |
| Test Case Input | Enter category name = “abca” |
| Test Case Expected Output | Null value |
| Test Case Steps | Step 1: Enter product ID  Step 2: Check database  Step 3: Return list Product |
|  |  |

|  |  |
| --- | --- |
| Test Case Number | 5.7 |
| Test Case Name | GetByCategoryTest1 |
| Test Case Description | This test case search Product by category |
| Preconditions | Logged in user |
| Test Case Input | Enter category name = “1234” |
| Test Case Expected Output | Null value |
| Test Case Steps | Step 1: Enter product ID  Step 2: Check database  Step 3: Return list Product |
|  |  |

|  |  |
| --- | --- |
| Test Case Number | 5.8 |
| Test Case Name | GetByCategoryTest1 |
| Test Case Description | This test case search Product by category |
| Preconditions | Logged in user |
| Test Case Input | Enter category name = “Freshs” |
| Test Case Expected Output | Null value |
| Test Case Steps | Step 1: Enter product ID  Step 2: Check database  Step 3: Return list Product |
|  |  |

|  |  |
| --- | --- |
| Test Case Number | 6.1 |
| Test Case Name | GetQuantity1 |
| Test Case Description | This test case get the quantity of the product |
| Preconditions | Logged in user |
| Test Case Input | ID Product = 1 |
| Test Case Expected Output | Return a product |
| Test Case Steps | Step 1: Enter product ID  Step 2: Check database  Step 3: Return a product |
|  |  |

|  |  |
| --- | --- |
| Test Case Number | 6.1 |
| Test Case Name | GetQuantity1 |
| Test Case Description | This test case get the quantity of the product |
| Preconditions | Logged in user |
| Test Case Input | ID Product = 1 |
| Test Case Expected Output | Return a product |
| Test Case Steps | Step 1: Enter product ID  Step 2: Check database  Step 3: Return a product |
|  |  |

|  |  |
| --- | --- |
| Test Case Number | 6.2 |
| Test Case Name | GetQuantity1 |
| Test Case Description | This test case get the quantity of the product |
| Preconditions | Logged in user |
| Test Case Input | ID Product = 15 |
| Test Case Expected Output | Return a product |
| Test Case Steps | Step 1: Enter product ID  Step 2: Check database  Step 3: Return a product |
|  |  |

|  |  |
| --- | --- |
| Test Case Number | 6.3 |
| Test Case Name | GetQuantity1 |
| Test Case Description | This test case get the quantity of the product |
| Preconditions | Logged in user |
| Test Case Input | ID Product = 20 |
| Test Case Expected Output | Return a product |
| Test Case Steps | Step 1: Enter product ID  Step 2: Check database  Step 3: Return a product |
|  |  |

|  |  |
| --- | --- |
| Test Case Number | 6.4 |
| Test Case Name | GetQuantity2 |
| Test Case Description | This test case get the quantity of the product |
| Preconditions | Logged in user |
| Test Case Input | ID Product = 0 |
| Test Case Expected Output | Null value |
| Test Case Steps | Step 1: Enter product ID  Step 2: Check database  Step 3: Return a product |
|  |  |

|  |  |
| --- | --- |
| Test Case Number | 6.5 |
| Test Case Name | GetQuantity2 |
| Test Case Description | This test case get the quantity of the product |
| Preconditions | Logged in user |
| Test Case Input | ID Product = -1 |
| Test Case Expected Output | Null value |
| Test Case Steps | Step 1: Enter product ID  Step 2: Check database  Step 3: Return a product |
|  |  |

|  |  |
| --- | --- |
| Test Case Number | 6.6 |
| Test Case Name | GetQuantity2 |
| Test Case Description | This test case get the quantity of the product |
| Preconditions | Logged in user |
| Test Case Input | ID Product = 999999999 |
| Test Case Expected Output | Null value |
| Test Case Steps | Step 1: Enter product ID  Step 2: Check database  Step 3: Return a product |
|  |  |

|  |  |
| --- | --- |
| Test Case Number | 7.1 |
| Test Case Name | GetByNo1 |
| Test Case Description | This test case search for an invoice by invoice number |
| Preconditions | Logged in user |
| Test Case Input | Input invoice no = 1001 |
| Test Case Expected Output | Return a Invoice |
| Test Case Steps | Step 1: Enter product ID  Step 2: Check database  Step 3: Return a Invoice |
|  |  |

|  |  |
| --- | --- |
| Test Case Number | 7.2 |
| Test Case Name | GetByNo1 |
| Test Case Description | This test case search for an invoice by invoice number |
| Preconditions | Logged in user |
| Test Case Input | Input invoice no = 1002 |
| Test Case Expected Output | Return a Invoice |
| Test Case Steps | Step 1: Enter product ID  Step 2: Check database  Step 3: Return a Invoice |
|  |  |

|  |  |
| --- | --- |
| Test Case Number | 7.1 |
| Test Case Name | GetByNo2 |
| Test Case Description | This test case search for an invoice by invoice number |
| Preconditions | Logged in user |
| Test Case Input | Input invoice no = -1 |
| Test Case Expected Output | Null value |
| Test Case Steps | Step 1: Enter product ID  Step 2: Check database  Step 3: Return a Invoice |
|  |  |

|  |  |
| --- | --- |
| Test Case Number | 7.1 |
| Test Case Name | GetByNo2 |
| Test Case Description | This test case search for an invoice by invoice number |
| Preconditions | Logged in user |
| Test Case Input | Input invoice no = -0 |
| Test Case Expected Output | Null value |
| Test Case Steps | Step 1: Enter product ID  Step 2: Check database  Step 3: Return a Invoice |
|  |  |

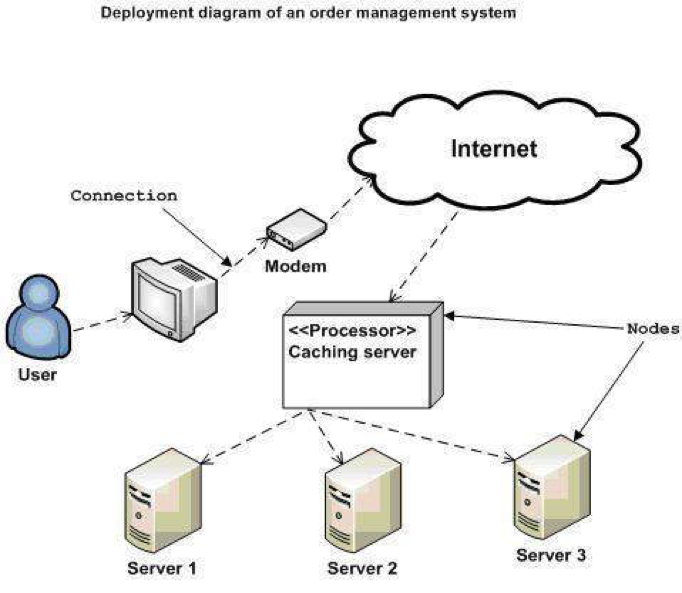
|  |  |
| --- | --- |
| Test Case Number | 7.1 |
| Test Case Name | GetByNo2 |
| Test Case Description | This test case search for an invoice by invoice number |
| Preconditions | Logged in user |
| Test Case Input | Input invoice no = 1000 |
| Test Case Expected Output | Null value |
| Test Case Steps | Step 1: Enter product ID  Step 2: Check database  Step 3: Return a Invoice |
|  |  |

# Task Assign (to each team member)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Analysis and design | | | | | |
| No | Task name | Description | Start Date | End Date | Member | Self assessment |
| 1 | Use-Case | Design the main functions of the software | 05/08 | 08/08 | Trần Văn Tiến, Mạc Hồng Phước, Vũ Văn Phúc |  |
| 2 | User-Case Description | Detailed description of the functions included in the software | 06/08 | 08/08 | Mạc Hồng Phước |  |
| 3 | Activity Diagram | Describe the operation of the functions | 10/08 | 12/08 | Vũ Văn Phúc,  Mạc Hồng Phước, Trần Văn Tiến |  |
| 4 | Sequence Diagrams | Describe in detail the transactions between layers |  |  | Trần Văn Tiến, Vũ Văn Phúc |  |
| 5 | Class Diagram | Describe the structure of the system, used to analyze the functions | 10/08 | 13/08 | Trần Văn Tiến |  |
| 6 | Database Design | Design the structure of the database | 14/08 | 25/08 | Trần Văn Tiến |  |
| 7 | Database Design Details | Detailed description of the entities in the database | 14/08 | 25/08 | Trần Văn Tiến |  |
| 8 | UI Design | Intuitive user interface design |  |  | Trần Văn Tiến |  |
| 9 | Add data to database | Create sample database | 01/09 | 10/09 | Mạc Hồng Phước,  Vũ Văn Phúc |  |
| 10 | Deploying the Persistence layer | Implement Persistence Layer Programming | 10/09 | 10/10 | Mạc Hồng Phước, Trần Văn Tiến |  |
| 11 | Implement all the DAL layer functionality | Implement Data Access Layer Layer Programming | 10/09 | 10/10 | Trần Văn Tiến |  |
| 12 | Implement all the BL layer functionality | Implement Business Layer Programming | 10/09 | 10/10 | Vũ Văn Phúc |  |
| 13 | Implement PL layer “Search By ID, Search By Name, Search By Category”functions | Programmatic implementation of functions and interfaces | 10/09 | 10/10 | Vũ Văn Phúc |  |
| 14 | Implement the "Create Invoice" function of the PL layer | Programmatic implementation of functions and interfaces | 10/09 | 10/10 | Trần Văn Tiến |  |
| 15 | Write a report |  | 07/10 | 12/10 | Trần Văn Tiến |  |

# Installation Instructions

1. Deployment Diagram



1. Installation steps
   * Database Install

- Go to the website and select the corresponding version: <https://dev.mysql.com/downloads/mysql/>

- Run the salessystem.sql file in the source file.

* + Server install
  + Application install

- Run file ConsoleApp.exe

# Appendix

Terms and abbreviations <if available>

<listing terms and abbreviations here>

References <if available >

<List of references here>

Some other issues <if available >

<Results, limitations, experiences, techniques, and other considerations when implementing a project>

# Document format

**Report corver:**

*The cover is printed in blue.*

*With the format as the first page of this document.*

**Paper Size:**

*The report is presented on A4 size paper (210 mm x 297 mm)*

**Top header**

*Left: Logo của VTC Academy*

*Right: Project Name*

*Font: Helvetica Neue (Light)*

*Font size: 12pt*

**Bottom header**

*Left: Class\_Name –Project\_Name*

*Right: Page\_Number*

*Font: Helvetica Neue (Light)*

*Font size: 12pt*

**Report Content:**

*Font: Helvetica Neue (Light)*

*Font size: 12pt*

*Minimum 20 pages*

**Page margin (for A4 size paper)**

*Top: 20 - 25 mm;*

*Bottom: 20 - 25 mm;*

*Left: 30 - 35 mm;*

*Right: 15 - 20 mm;*