

**PROJECT REPORT**

# Coffee Shop Management

|  |  |
| --- | --- |
| **Semester:** | Programming Fundamentals |
| **Class:** | WD1111 |
| **Group:** | Coffee Shop |
| **Instructor** | Nguyen Dinh Cuong |
| **Team members:** | Nguyen Ngoc Duc  Vu Cong Nguyen |

# Index

Coffee Shop Management 1

Index 2

I. Project introduction 3

II. Analyze System Requirements 4

III. Design Details 25

IV. Test 33

2. Task Assign (to each team member) 40

3. Installation Instructions 41

# Project introduction

The system will provide some basic features including:

* Login user account.
* Admin:
  + Add, update, change status products
  + Add, update, change status staff
  + Add, update, view products in category
  + View dashboard
* Seller:
  + Create, change status, handle order
* Bartender:
  + Receive order
  + Report product status instock
* Guest:
  + View product details
  + Add products to cart
  + Create new account
  + Forgot password

Which includes additional features such as view order list, view product list, revenue statistic.

1. Proposed System

* Create a program for sales staff to create orders, update order status, or cancel order, and handle errorneos orders.
* Guests can be orders add products to their shopping carts, and directly pay for orders on the website.
* Bartender handle pending order and notify ready order, as well as report the availablility status of products being sold, whether they are instock or out of stock.

1. The scope of the project to be applied

* Make for actor are staff, guest.

1. System Name

* Coffee shop management.

1. Deployment Environment

* Windows, Linux, Mac.

1. Development Tools

* Visual Studio Code
* Draw.io Diagram
* MAMP

1. Customer Requirements (System features)

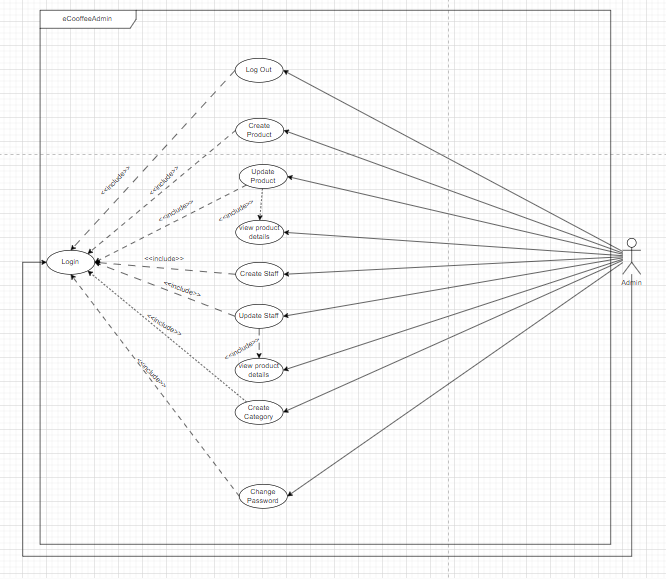
* Staff :
  + User login: user must login to use the system.
  + Create order: staff can create order according to customer’s request.
  + Update order: staff can update order like as change product, remove product, add product to order, handle error order.
  + Payment: used to check complete order and pay order.
  + Check out: staff can check out.
* Guest:
  + User login: guest may choose to login ornot to use the system
  + Add to cart: guest can be orders add products to their shopping carts.
  + Check out: Guest can create and check out order directly on the website
  + Payment: guest can make online payments for orders though VNPay

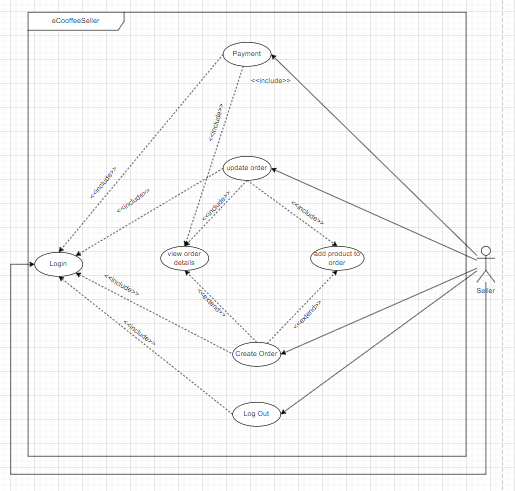
# Analyze System Requirements

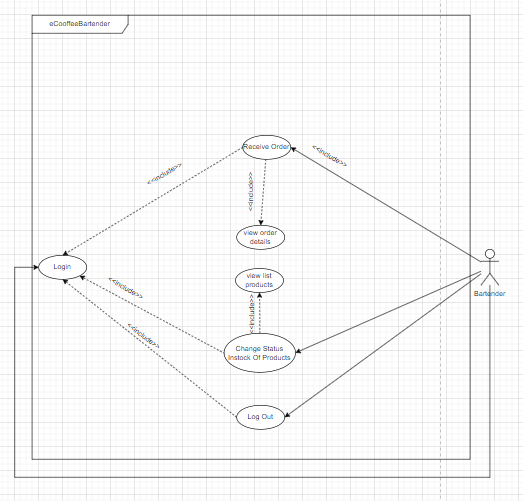
This system is designed to help café manage orders easily. It can provide staff with additional information about the coffee sold in the shop, including prices, names, and statuses. The main functions of this system for staff are to assist them in creating orders as per customer requests, allowing them to view unpaid orders and confirm successful payments if they have received sufficient payment from the customer, receiving and processing created orders, notifying the status of products in stock, handling erroneous orders, and canceling orders. It also allows placing orders when customers change their minds or temporarily suspend payments when necessary.

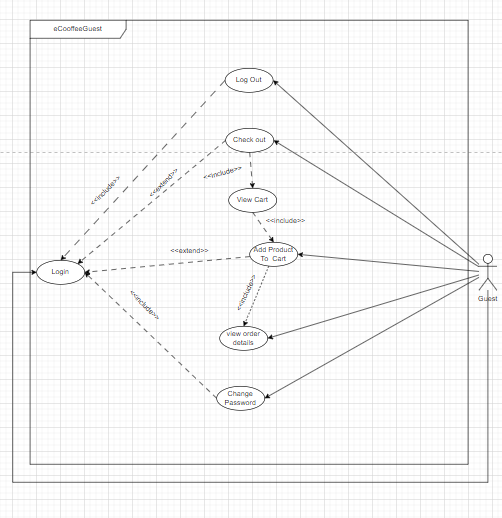
1. Use Case
   1. Use Case

------------------------------------









* 1. Login description

|  |  |
| --- | --- |
| Use Case Name | Login |
| Use Case ID | UC01\_Login |
| Description | Used to access the program and use system. |
| Actor | Staff |
| Organizational Benefits | The feature was created to define the features and data that staff can access in the program. |
| Triggers | Login will be automatically activated as soon as we start the program. |
| Preconditions | Must enter the correct username and password, database must be connected, account must be made available. |
| Postconditions | Success: Staff have logged into the system, the main menu of the program must be displayed.  Fail: Staff haven’t logged into the system. |
| Main Course | Step 1: Input username and password.  Step 2: Verify login.  Step 3: Show main menu. |
| Alternate Courses | AC 1: Username and password exists.  Step 1: Show successful login message.  Step 2: Show main menu |
| Exceptions | EX 1: User enter wrong username, password  Step 1: Display message error.  Step 2: Show the login screen again.  Step 3: Re-enter request.  EX 2: Enter invalid user name or password (special character or invalid structure)  Step 1: Display message error.  Step 2: Show the login screen again.  Step 3: Re-enter request.  EX 3: Disconnect to database  Step 1: Display “Database is disconnected, please check your internet.”  Step 2: Request to check connection. |

* 1. Add product to order description

|  |  |
| --- | --- |
| Use Case Name | Add product to order |
| Use Case ID | UC02\_AddProduct |
| Description | Used to add product to order |
| Actor | Staff |
| Organizational Benefits | You can check valid properties of product (id, quantity, instock) and add product to order if product is valid. |
| Triggers | This feature will be activated in create order feature, update order feature. |
| Preconditions | The functions to create order or update order is selected and a product must be selected. |
| Postconditions | Success: Add product complete  Fail: Add product not complete. |
| Main Course | 1. Input product id, size and quantity you want to order. 2. Valid data. 3. Add product complete. |
| Alternate Courses | AC 1: Add product or update complete.  Step 1: Show success message.  AC 2: Exit (choose 0)  Step 1: Show main menu if there are no products in order or show order details if products already in order. |
| Exceptions | EX 1: Invalid product id (input special character or alphanumeric character, id not exits in menu)  Step 1: Display message error.  Step 2: Request re-entry.  EX 2: Invalid quantity (input special character or alphanumeric character).  Step 1: Display message error.  Step 2: Request re-entry.  EX 3: The selected product does not have the required quantity.  Step 1: Show message error.  Step 2: Request to re-entry quantity.  EX 4: Disconnect to database.  Step 1: Display “Database is disconnected, please check your internet.”  Step 2: Request to check connection. |

* 1. Create order description

|  |  |
| --- | --- |
| Use Case Name | Create Order |
| Use Case ID | UC03\_CreateOrder |
| Description | Used to create order to use at the shop or takeaway. |
| Actor | Staff |
| Organizational Benefits | You can create one or more orders according to customer’s request. |
| Triggers | This feature will be activated when staff use choose it in main menu. |
| Preconditions | Must have at least one product previously created. |
| Postconditions | Success: Create order complete.  Fail: Create order not complete. |
| Main Course | 1. Choose use at the shop order take away. 2. If choose use at shop, show table management and choose table ID. 3. Show menu and request to add product to order. 4. Valid data. 5. Confirm request. 6. Create order complete. |
| Alternate Courses | AC 1: Continue add product to order.  Step 1: After adding a product to order, display message “Do you want to continue add product to order?”.  Step 2: Choose Yes.  Step 3: Show menu and ask to add again. |
| Exceptions | EX 1: Invalid choice (input special character or alphanumeric character or other choice in menu)  Step 1: Display message error.  Step 2: Request re-entry.  EX 2: Cancel order before confirming.  Step 1: After selecting the product and quantity, show message “Do you want to create the order?”.  Step 2: Choose No.  Step 3: Show canceled order and delete order.  EX 3: Disconnect to database.  1. Step 1: Display “Database is disconnected, please check your internet.”  2. Step 2: Request to check connection. |

* 1. Update order description

|  |  |
| --- | --- |
| Use Case Name | Update order |
| Use Case ID | UC04\_UpdateOrder |
| Description | Used to update order. |
| Actor | Staff |
| Organizational Benefits | You can update order information. |
| Triggers | Login to program to use this feature. |
| Preconditions | The order must be exist. |
| Postconditions | Success: Update order complete.  Fail: Update order not complete. |
| Main Course | MC 1: Add product to current order.   1. Show menu and ask to choose product you want to add to order. 2. Add product to order, show complete message and ask to continue add.   MC 2: Change product information.   1. Show list product in order. 2. Choose product in order. 3. Change product order, show complete message and ask to continue update.   MC 3: Remove product.   1. Show list product in order and ask to choose product. 2. Remove product, show complete message and ask to continue remove.   MC 4: Confirm product in order.   1. Show list product in order and ask to choose product. 2. Confirm product and ask to continue confirm.   MC 5: Change table.   1. Show list table and ask to choose table. 2. Change table, show complete message and ask to continue change. |
| Alternate Courses | AC 1: Show order details.   1. Select the order you want to view details in list all order. 2. If order valid, Show all order information. |
| Exceptions | EX 1: The select order does not exist.  Step 1: Display “Order does not exist, Please select the order form the list. ”  Step 2: Request to choose order again.  EX 2: Disconnect to database.  Step 1: Display “Database is disconnected, please check your internet.”  Step 2: Request to check connection.  EX 3: Invalid choice (input special character or alphanumeric character or other choice in menu)  Step 1: Display message error.  Step 2: Request re-entry.  EX 4: Cancel update order before confirming.  Step 1: Show message “Do you want to update the order?”.  Step 2: Choose No.  Step 3: Show canceled update order and return order. |

* 1. Payment description

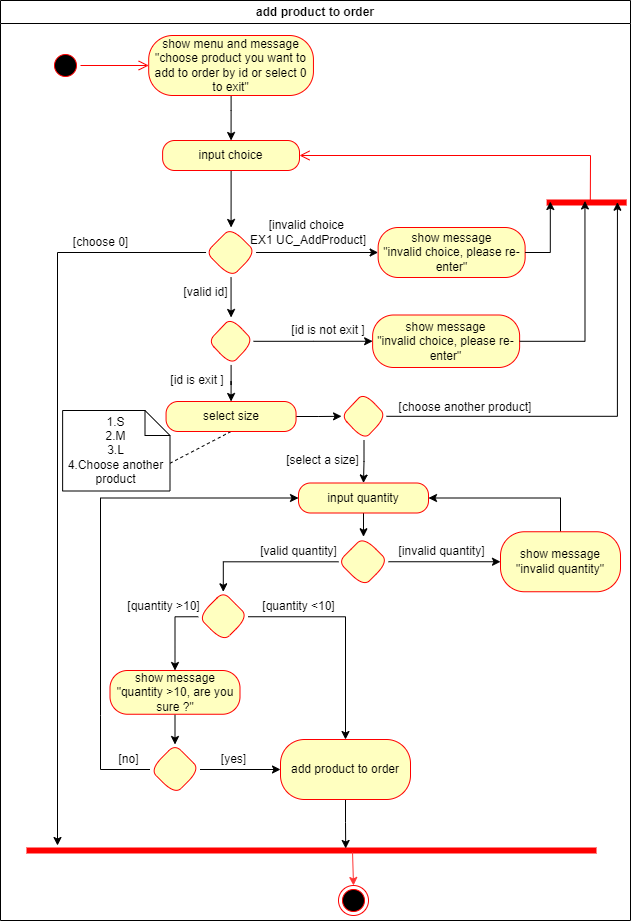
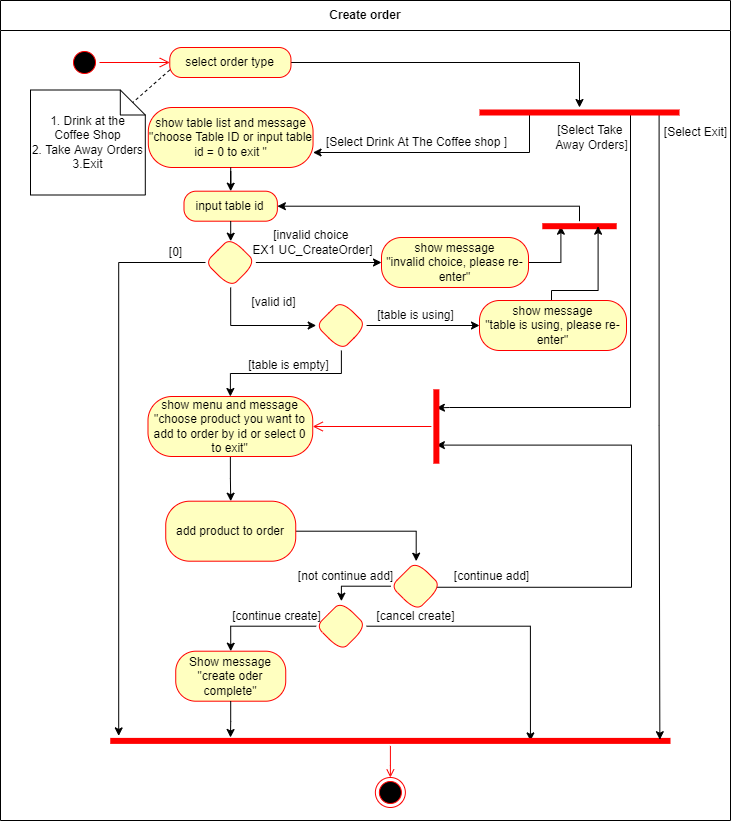
|  |  |
| --- | --- |
| Use Case Name | Payment |
| Use Case ID | UC05\_Payment |
| Description | Used to payment confirmation. |
| Actor | Staff |
| Organizational Benefits | You can confirm payment. |
| Triggers | Login to program to use this feature. |
| Preconditions | The order must be exist. |
| Postconditions | Success: Confirm payment complete.  Fail: Confirm payment not complete . |
| Main Course | 1. Select the order you want to payment in list all order. 2. If order valid, Show order information and total amount . 3. Confirm payment |
| Alternate Courses | AC1: Confirm payment.   1. Request confirm. 2. Confirm complete.   AC2: Not confirm payment.   1. Request confirm. 2. Choose 0. 3. Back to list order.   AC3: Product not complete.   1. Request ask “Do you want delete unfinished product and continue confirm payment”. 2. Choose Yes. 3. Delete unfinished product and confirm. |
| Exceptions | EX 1: The select order does not exist.  Step 1: Display “Order does not exist, Please select the order form the list. ”  Step 2: Request to choose order again.  EX 2: Disconnect to database.  Step 1: Display “Database is disconnected, please check your internet.”  Step 2: Request to check connection.  EX 3: Invalid choice (input special character or alphanumeric character or other choice in menu)  Step 1: Display message error.  Step 2: Request re-entry. |

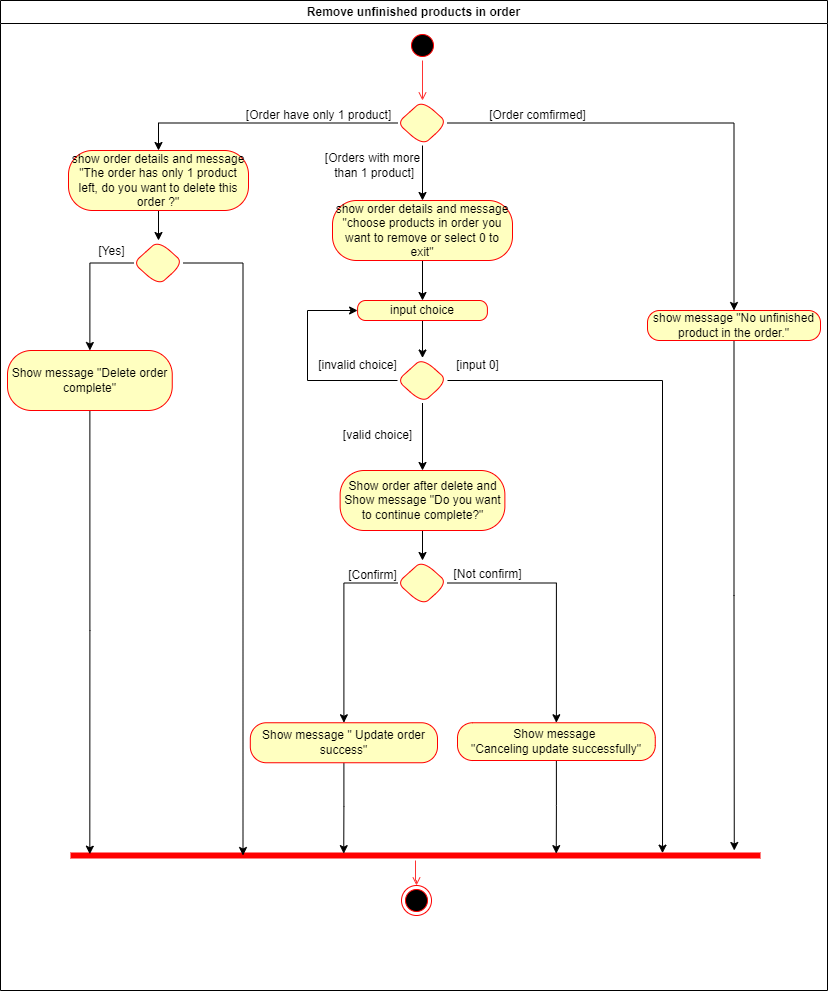
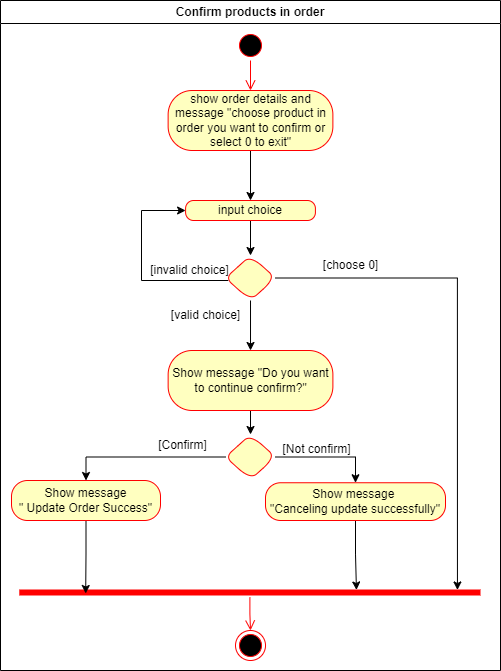
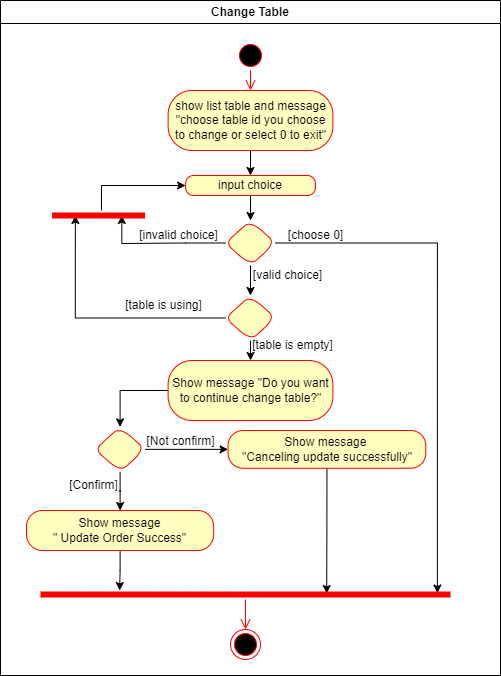
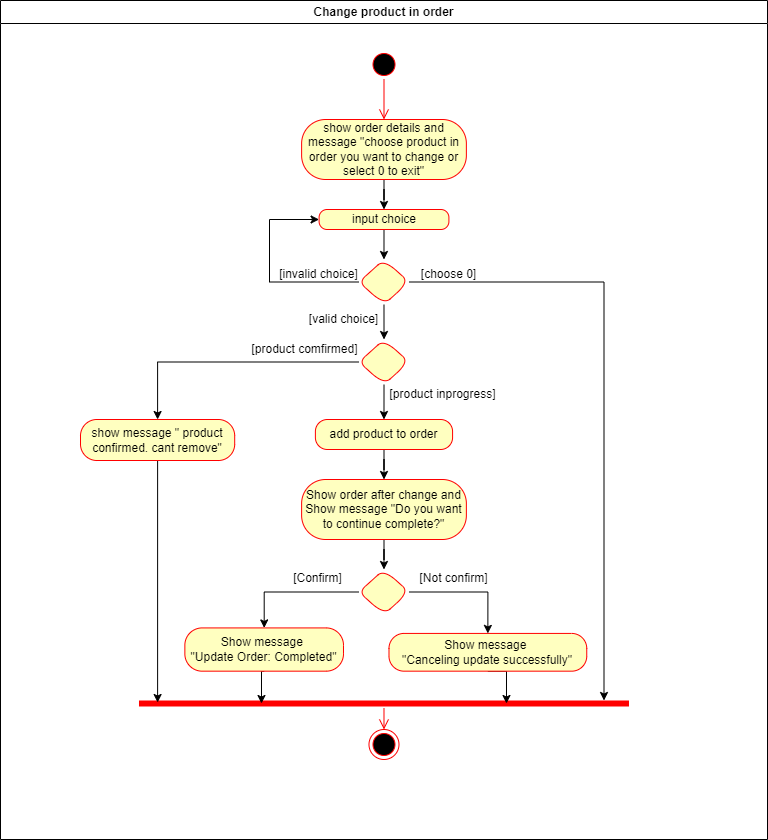
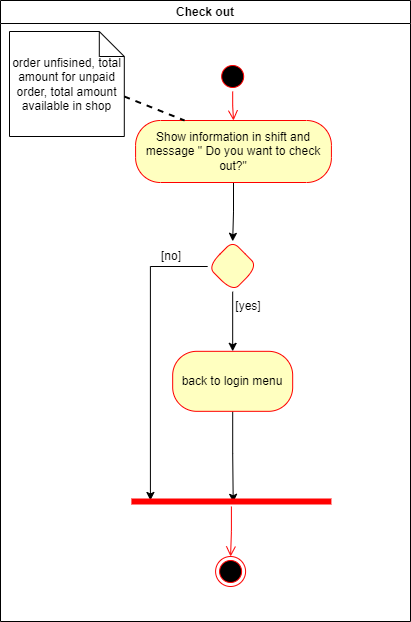
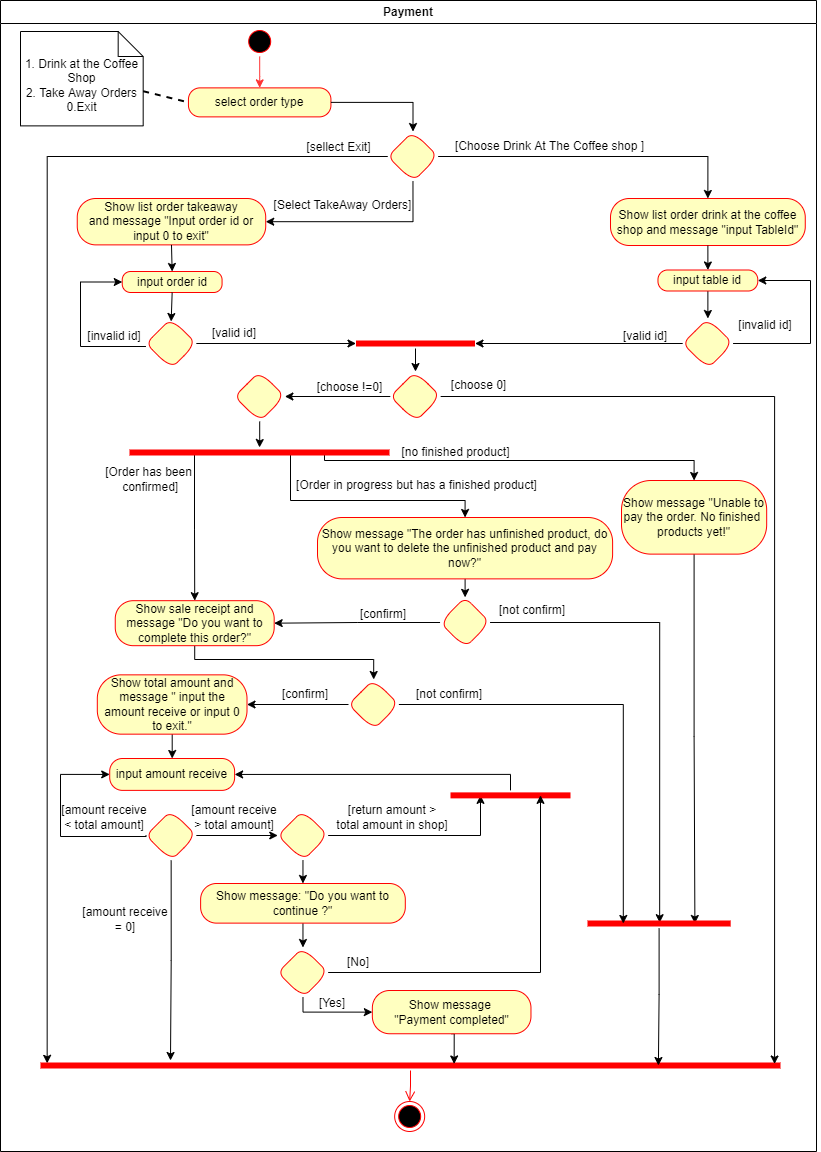
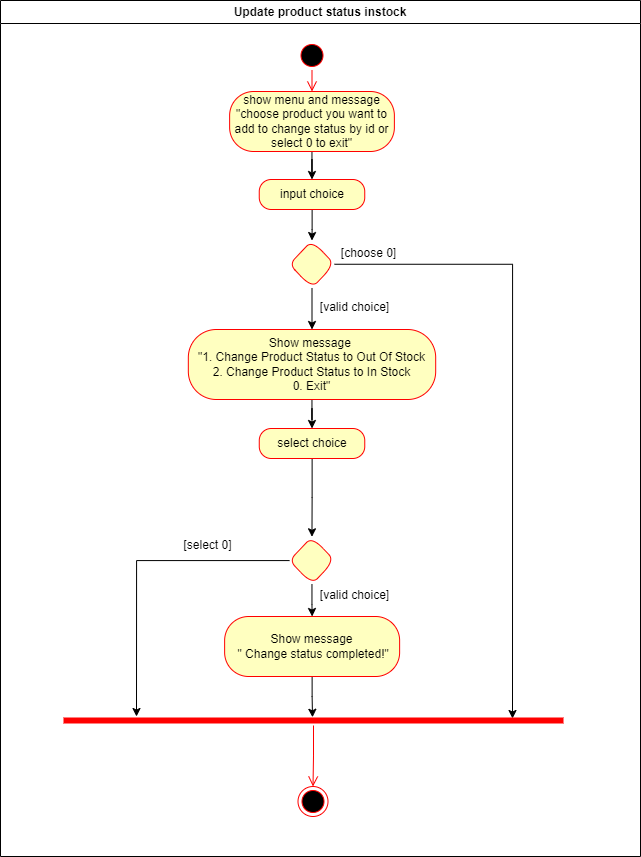
* 1. Check Out description

|  |  |
| --- | --- |
| Use Case Name | Check Out |
| Use Case ID | UC06\_CheckOut |
| Description | Used to check out. |
| Actor | Staff |
| Organizational Benefits | You can check out. |
| Triggers | Login to program to use this feature. |
| Preconditions | Must have all order complete |
| Postconditions | Success: Check out complete.  Fail: Check out not complete. |
| Main Course | 1. Show all order unconfirm in shift and total amount in safe before check out. 2. Request to check out. 3. Check out and back to login menu. |
| Alternate Courses | AC 1: Check out.   1. Show total amount in the safe. 2. Back to login menu.   AC 2: Exit   1. Back to main menu. |
| Exceptions | EX 1: Invalid choice (input special character or alphanumeric character or other choice in menu)  Step 1: Display message error.  Step 2: Request re-entry. |

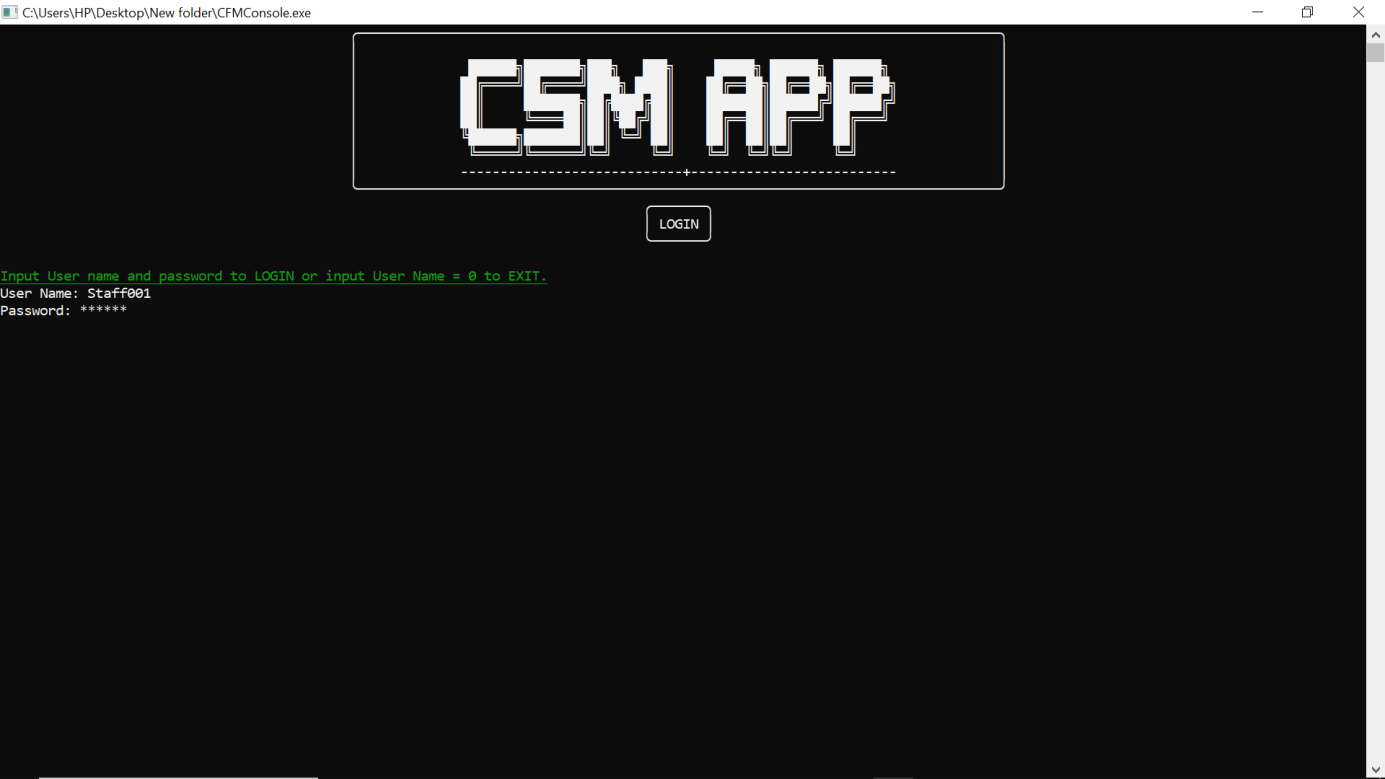
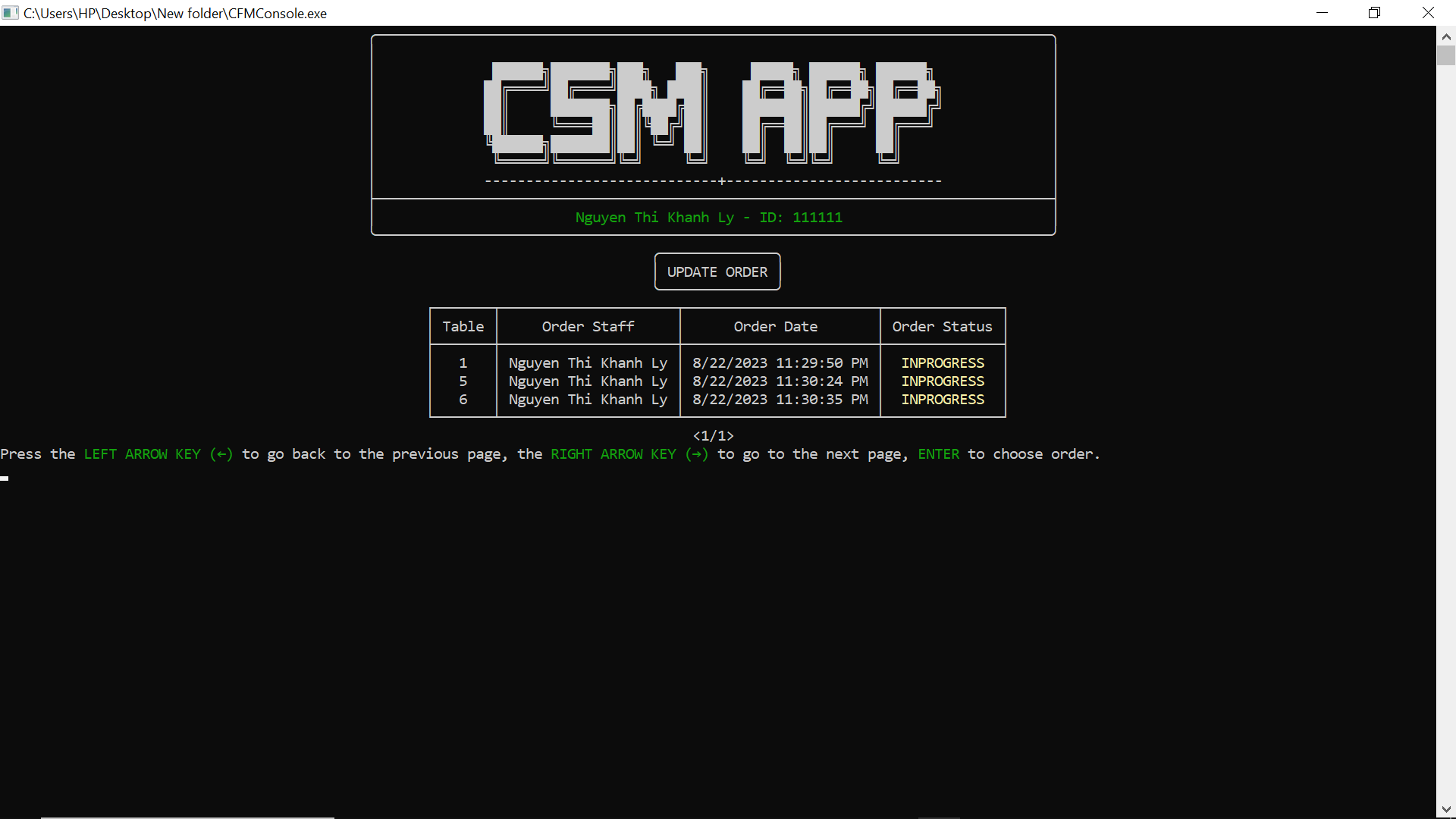
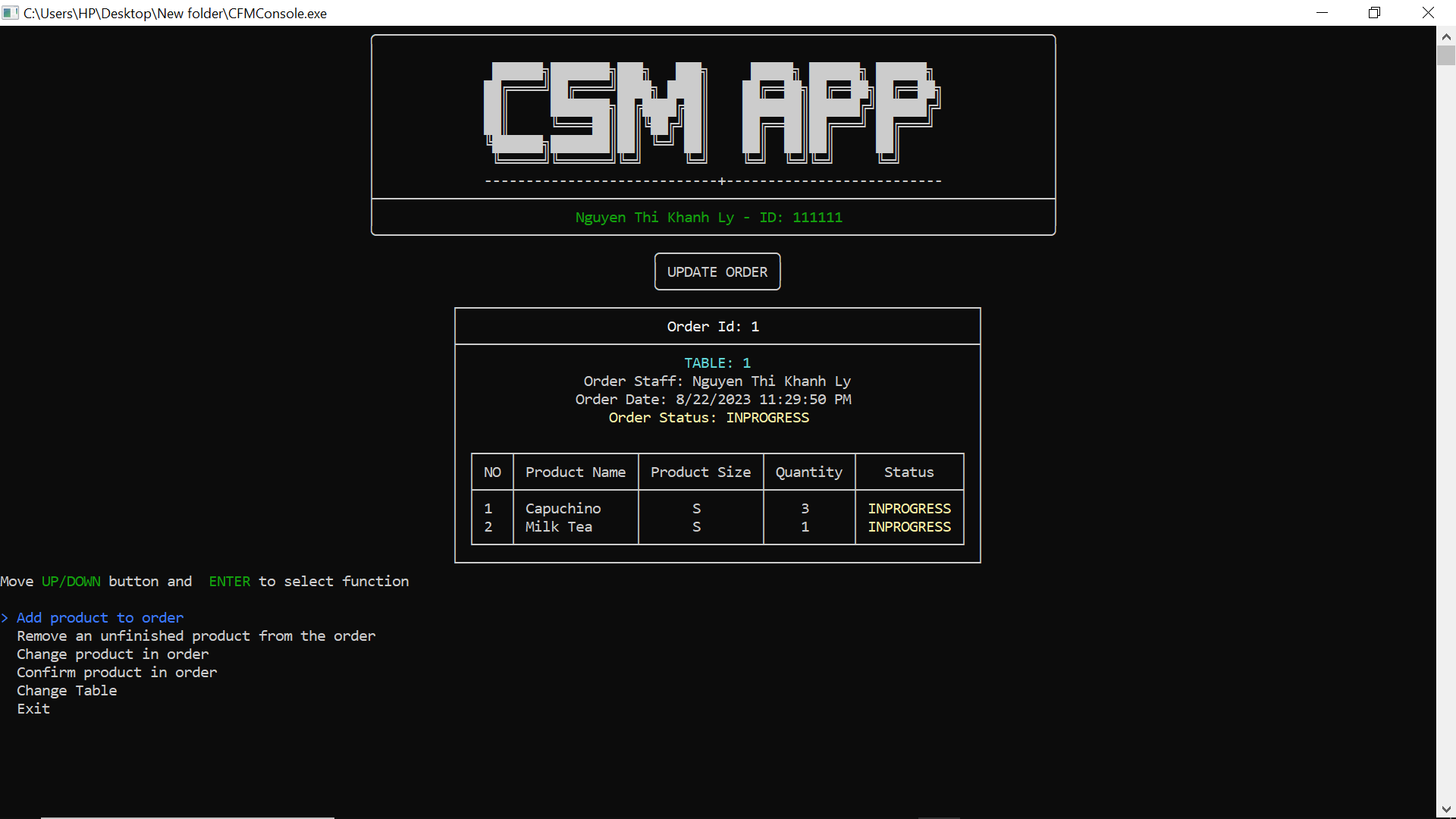
* 1. Update Product Status Instock description

|  |  |
| --- | --- |
| Use Case Name | Update Product Status Instock |
| Use Case ID | UC07\_ UpdateProductStatusInstock |
| Description | Used to update status of product. |
| Actor | Staff |
| Organizational Benefits | You can update product status instock. |
| Triggers | Login to program to use this feature. |
| Preconditions | Must have products. |
| Postconditions | Success: Change status completed!.  Fail: Back to menu. |
| Main Course | 1. Show all products in menu. 2. Input ID product you want to update. 3. Select status you want to change and confirm. |
| Alternate Courses | AC 1: Exit   1. Back to main menu. |
| Exceptions | EX 1: Invalid choice (input special character or alphanumeric character or other choice in menu)  Step 1: Display message error.  Step 2: Request re-entry.  EX 2: Disconnect to database.  Step 1: Display “Database is disconnected, please check your internet.”  Step 2: Request to check connection. |

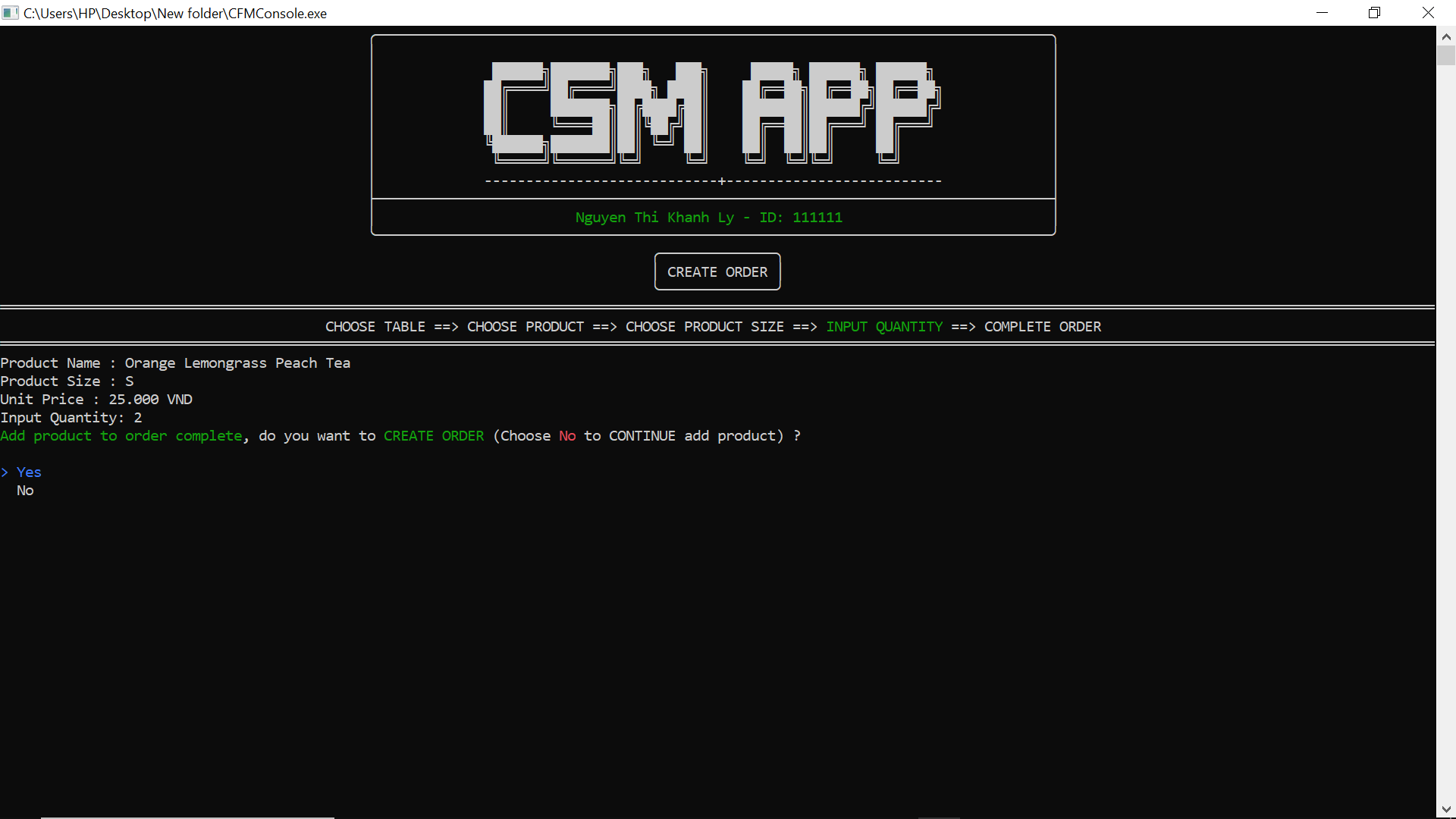
1. Activity Diagram: A diagram of a work flow

   Description automatically generated

# Design Details

1. UI Design
   1. Login 
   2. Main Menu 
   3. Show Order List 
   4. Menu Update Order 
   5. Menu Product



* 1. Create Order



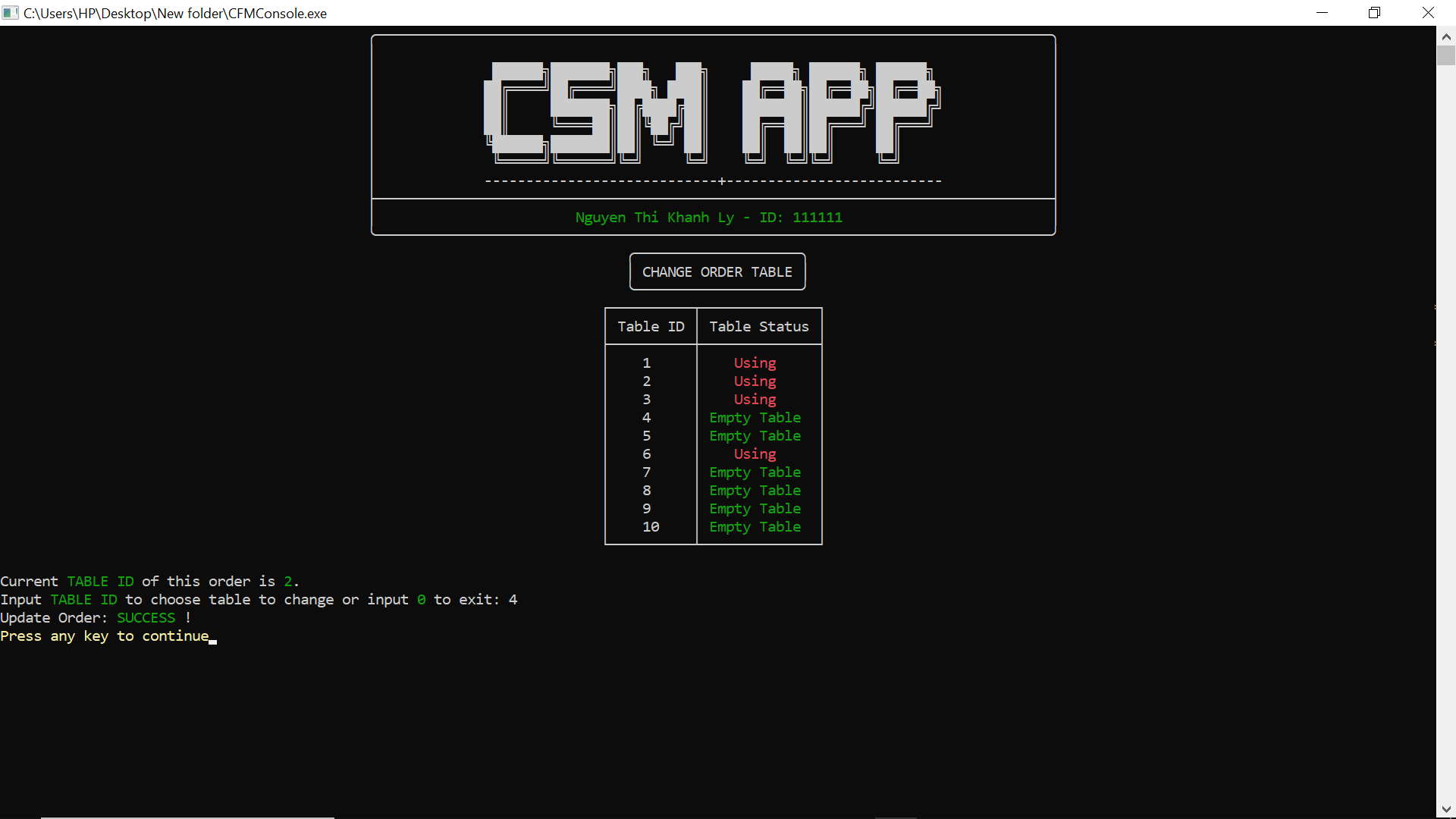
* 1. Menu Status Product 
  2. Update Order

1.8.1. Add product to order 

1.8.2. Remove an unfinished product 

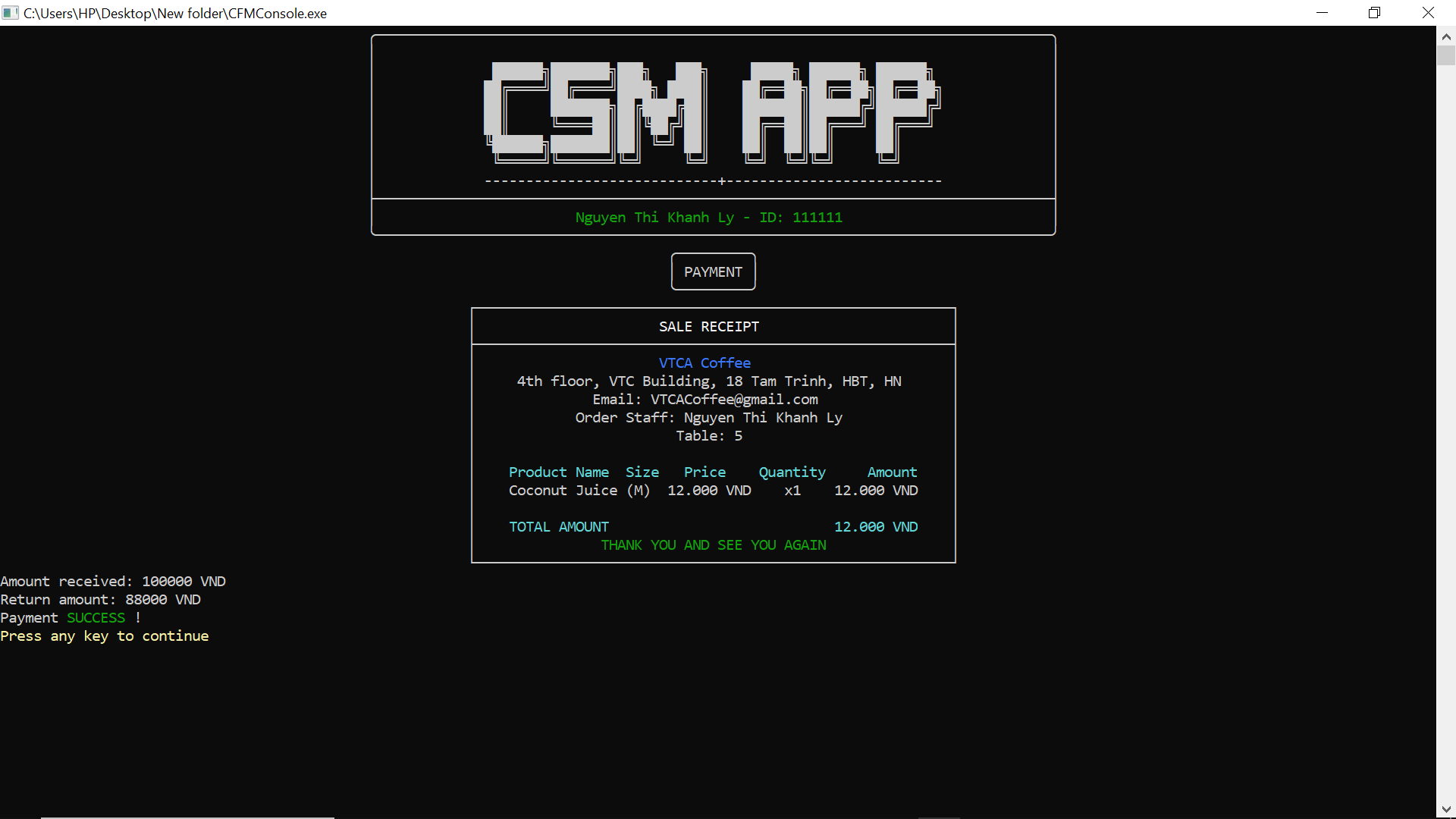
1.8.3. Change product in order 

1.8.4. Confirm product in order 

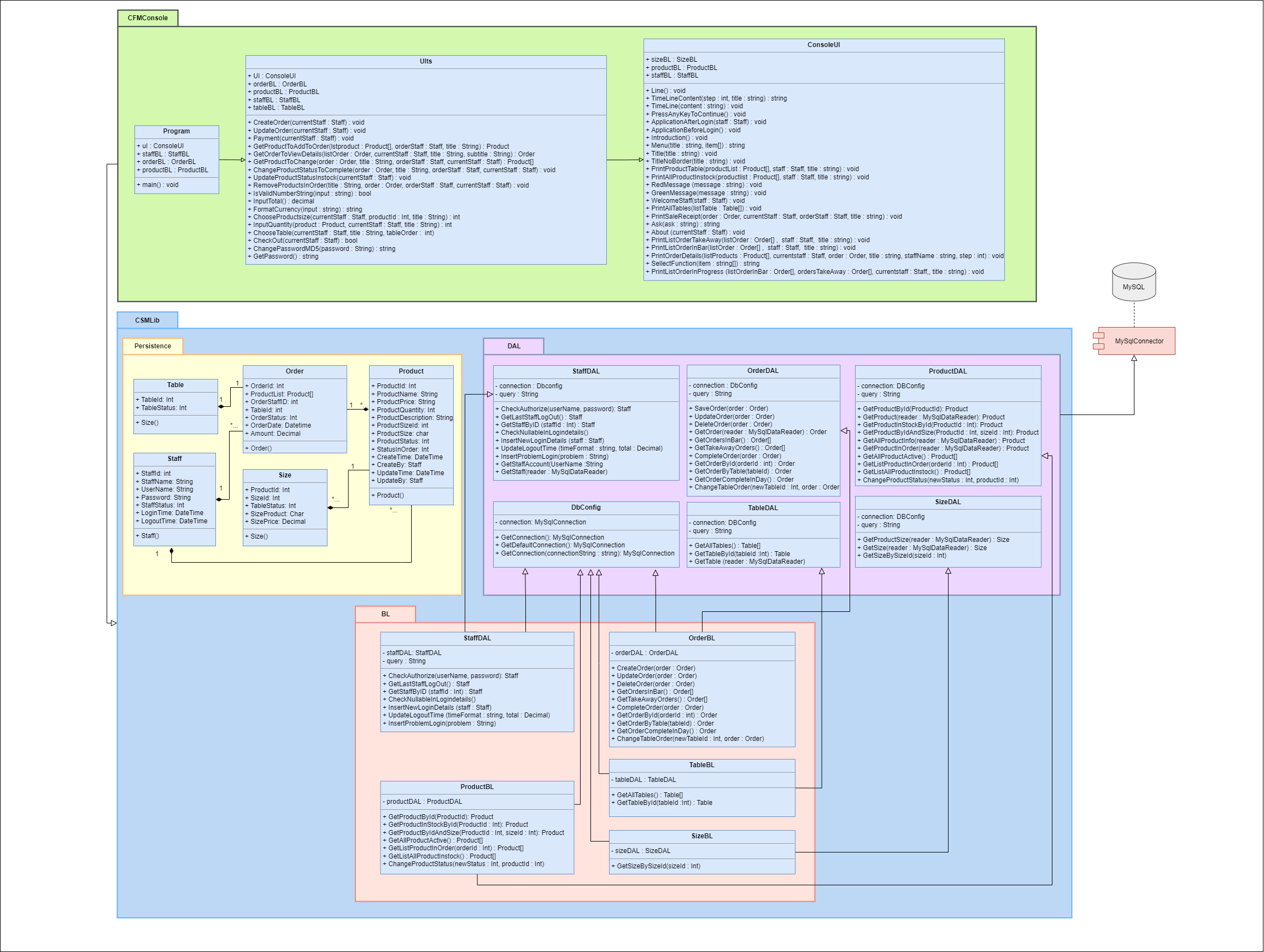
1.8.5. Change Table 

* 1. Payment

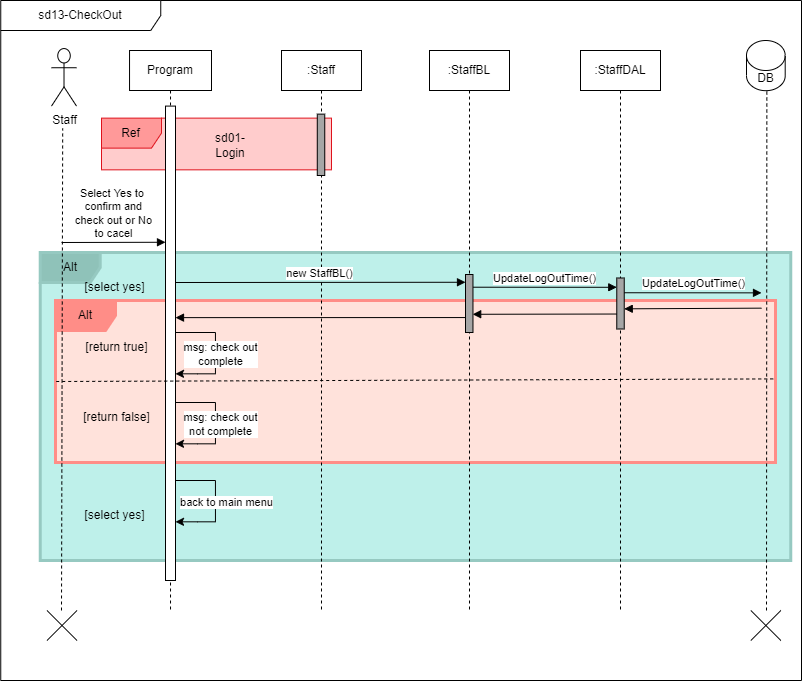
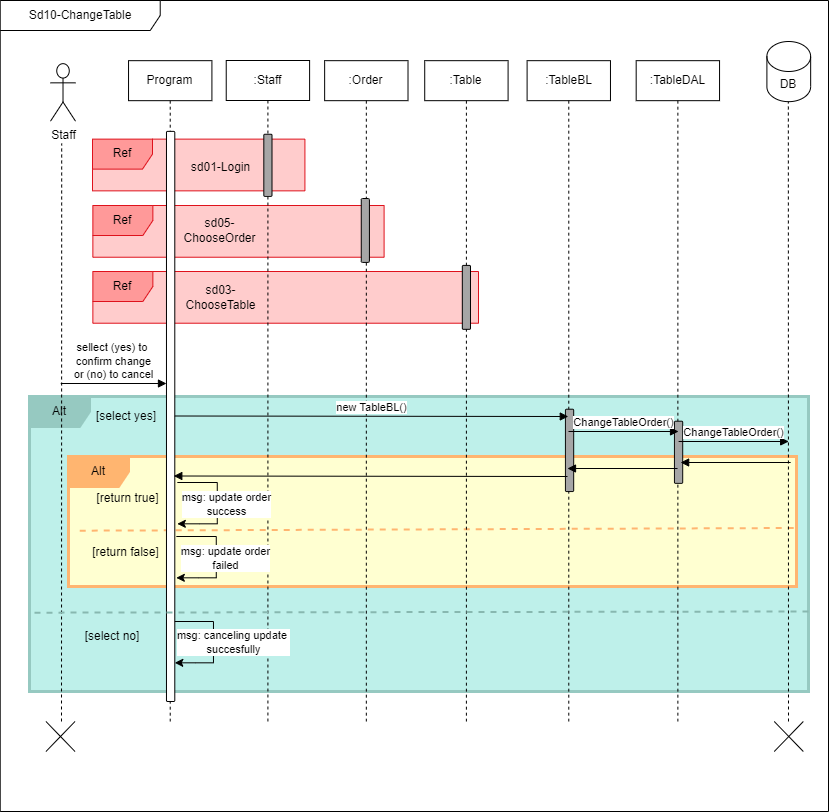
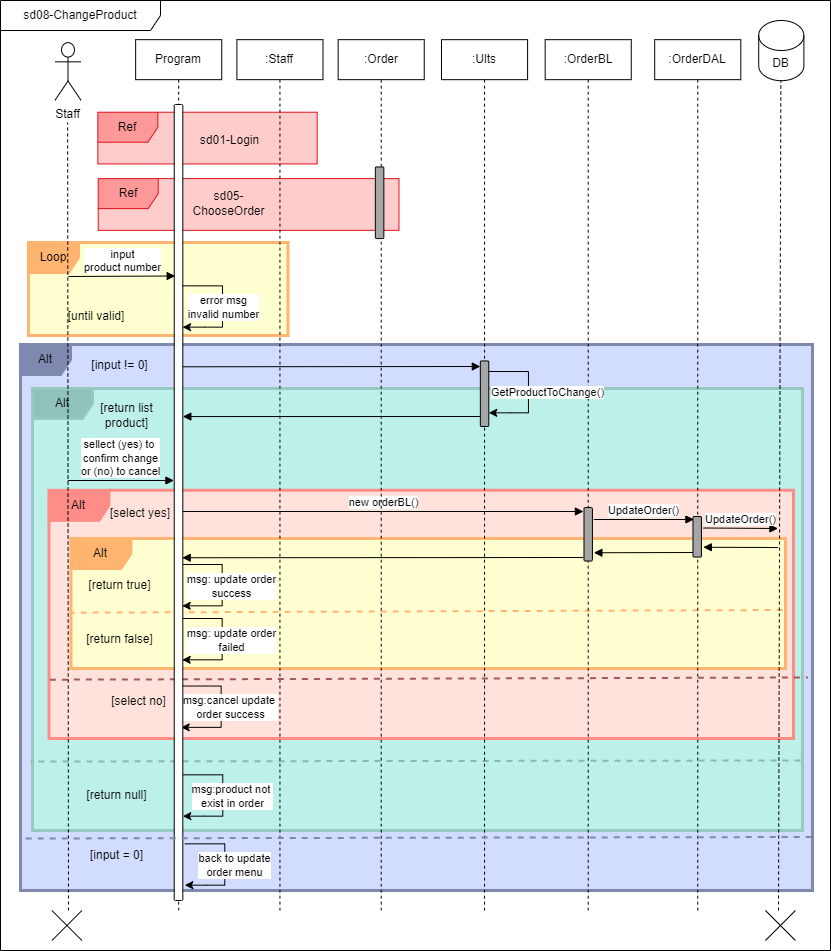
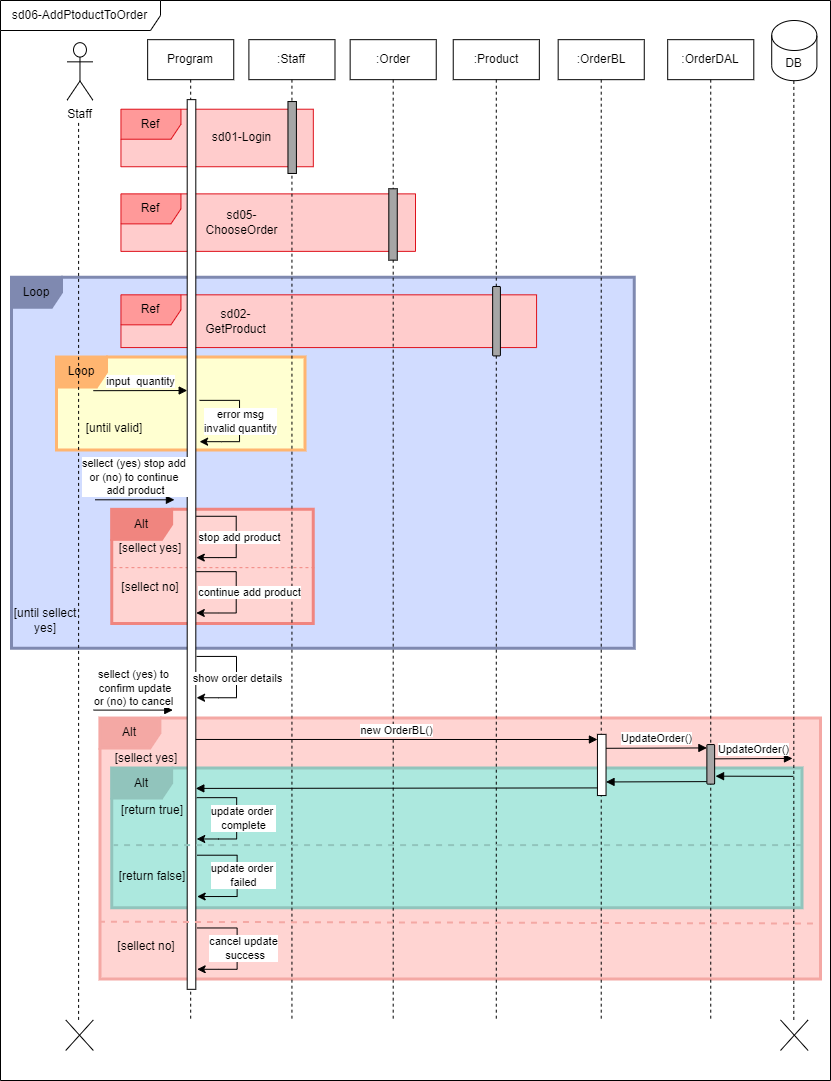
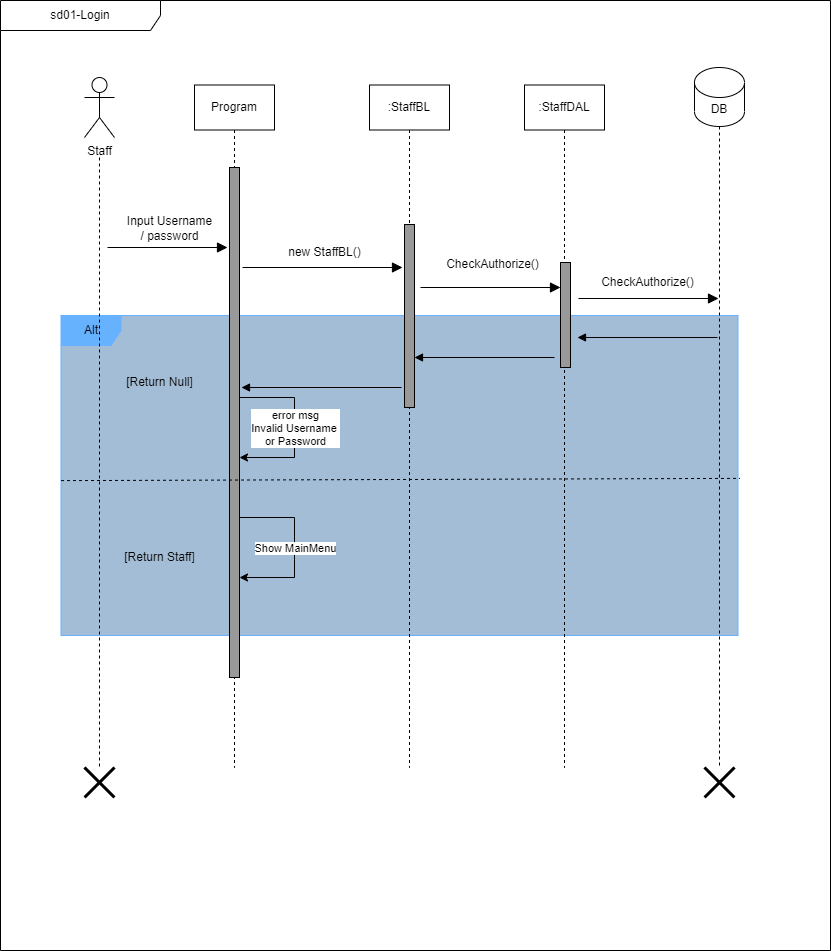


* 1. Invoice 

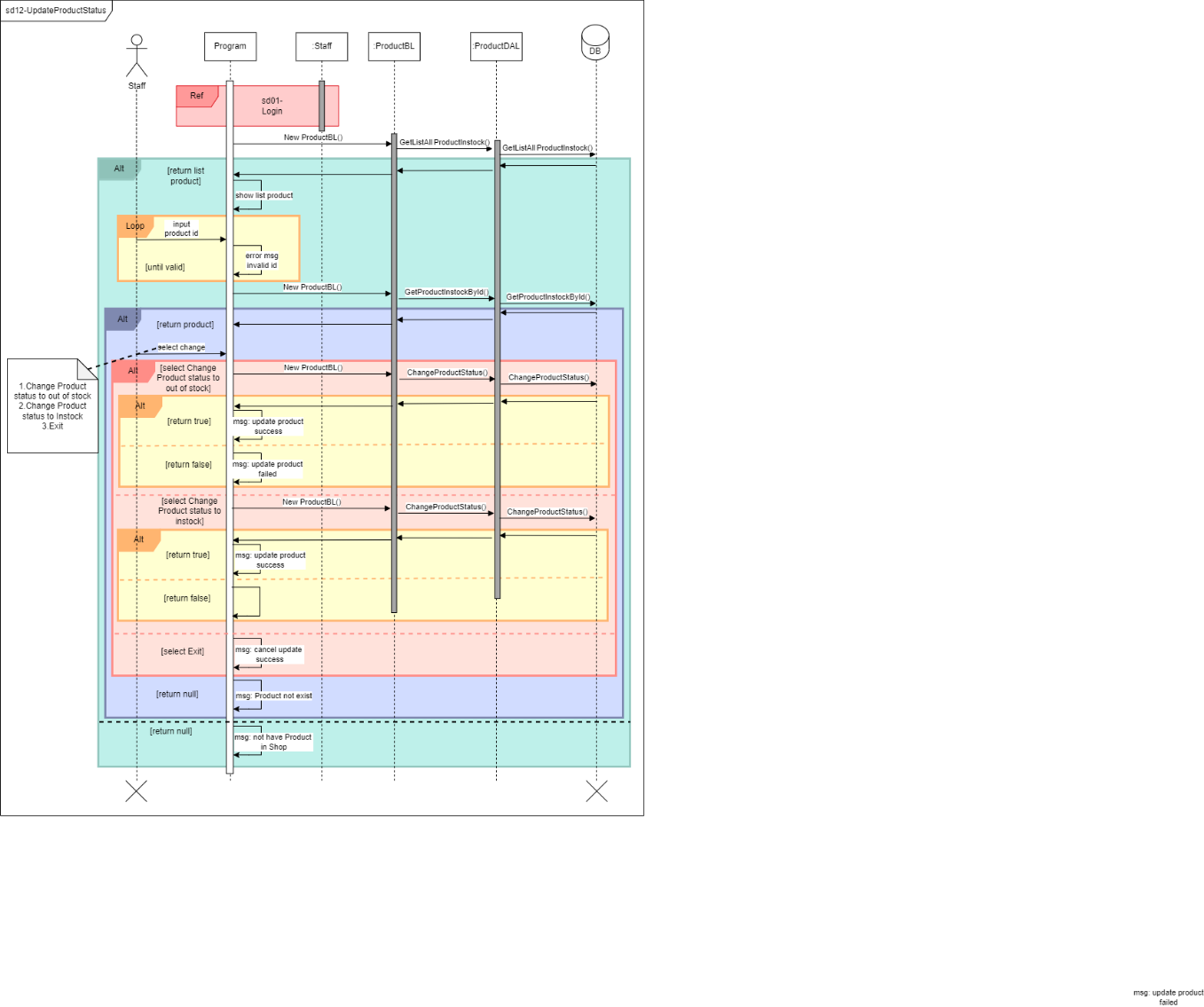
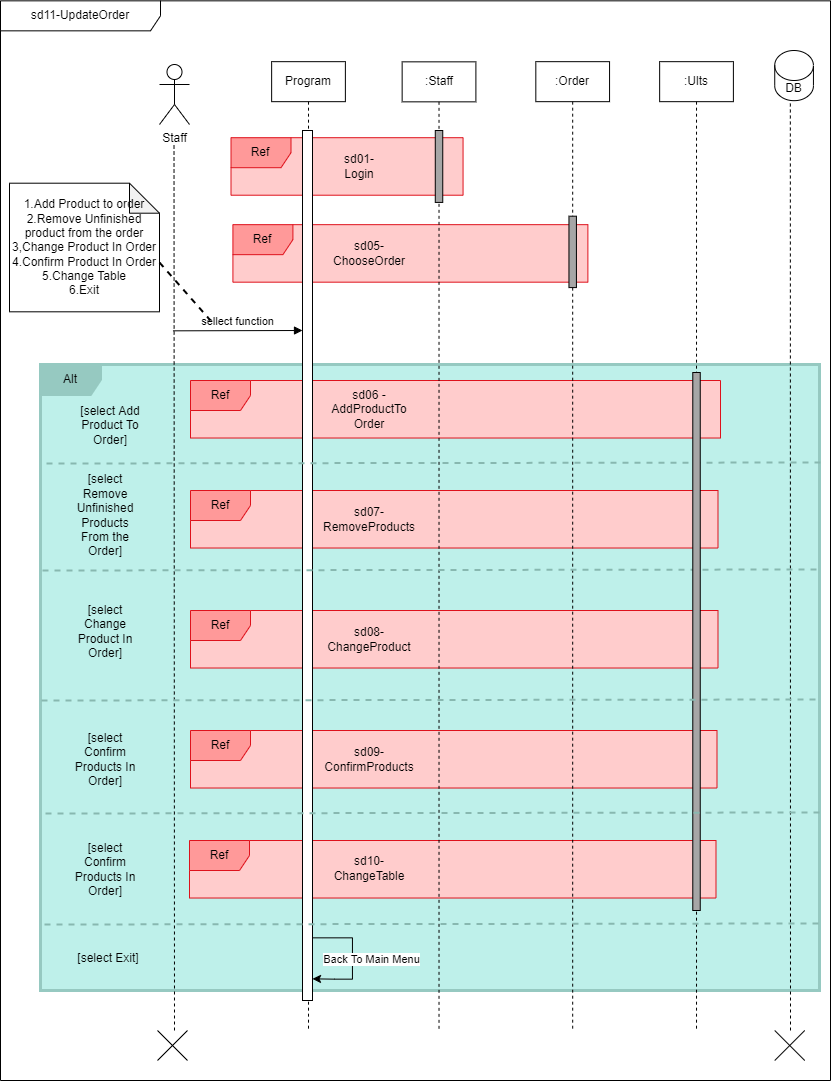
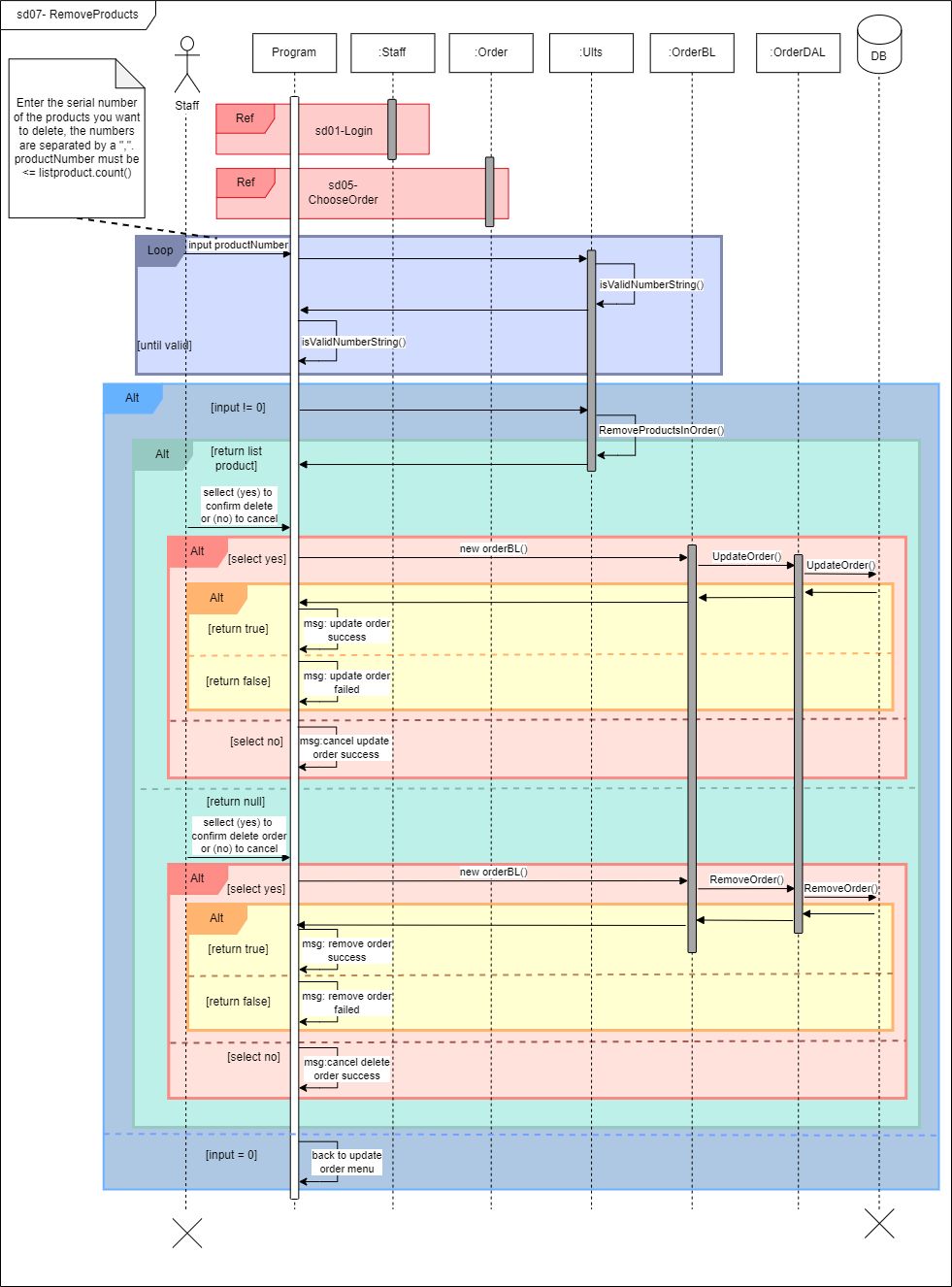
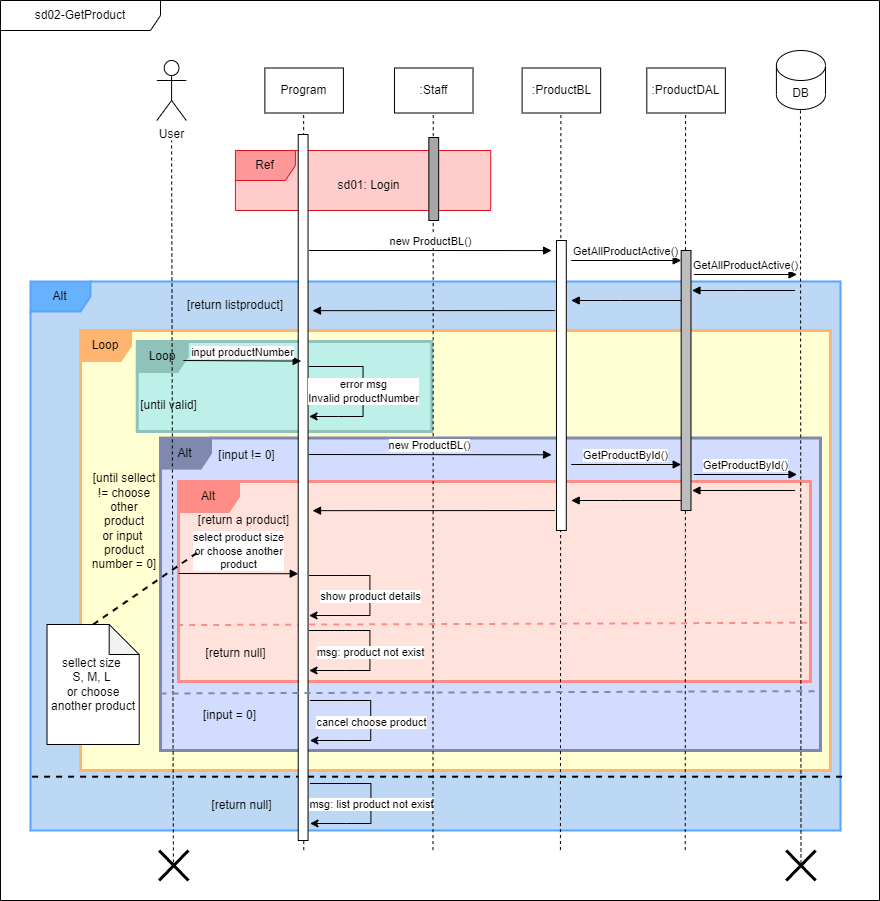
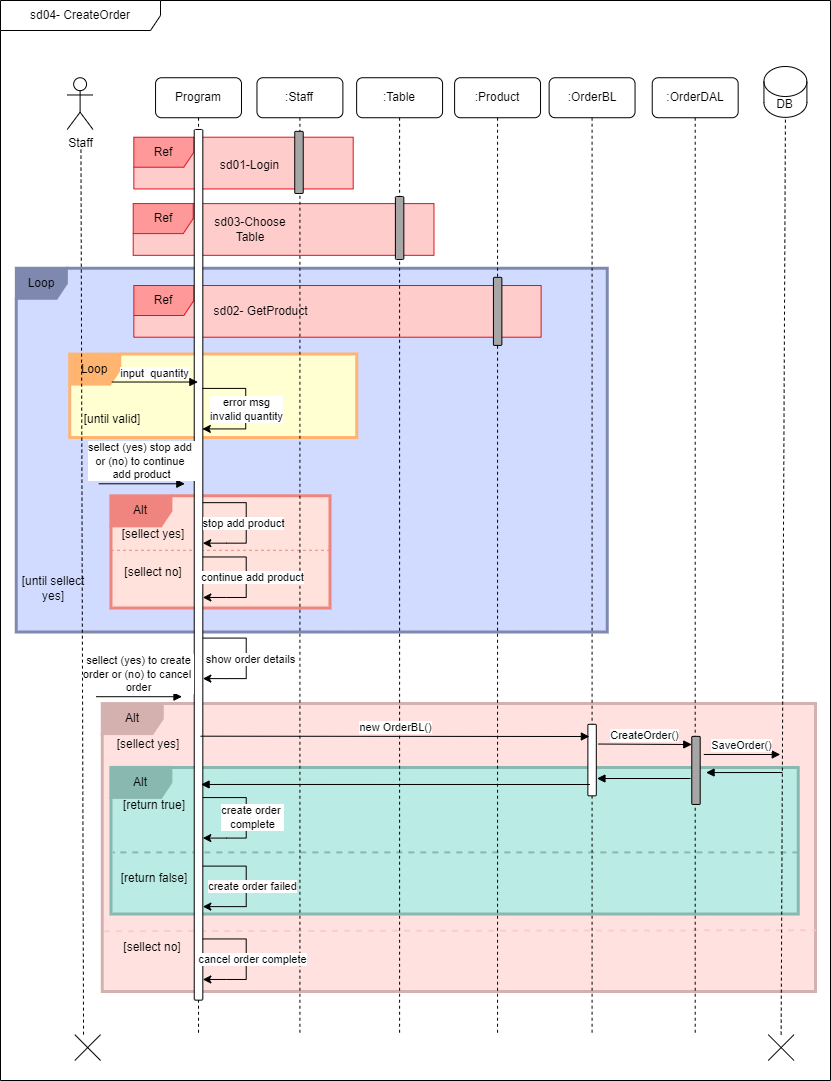
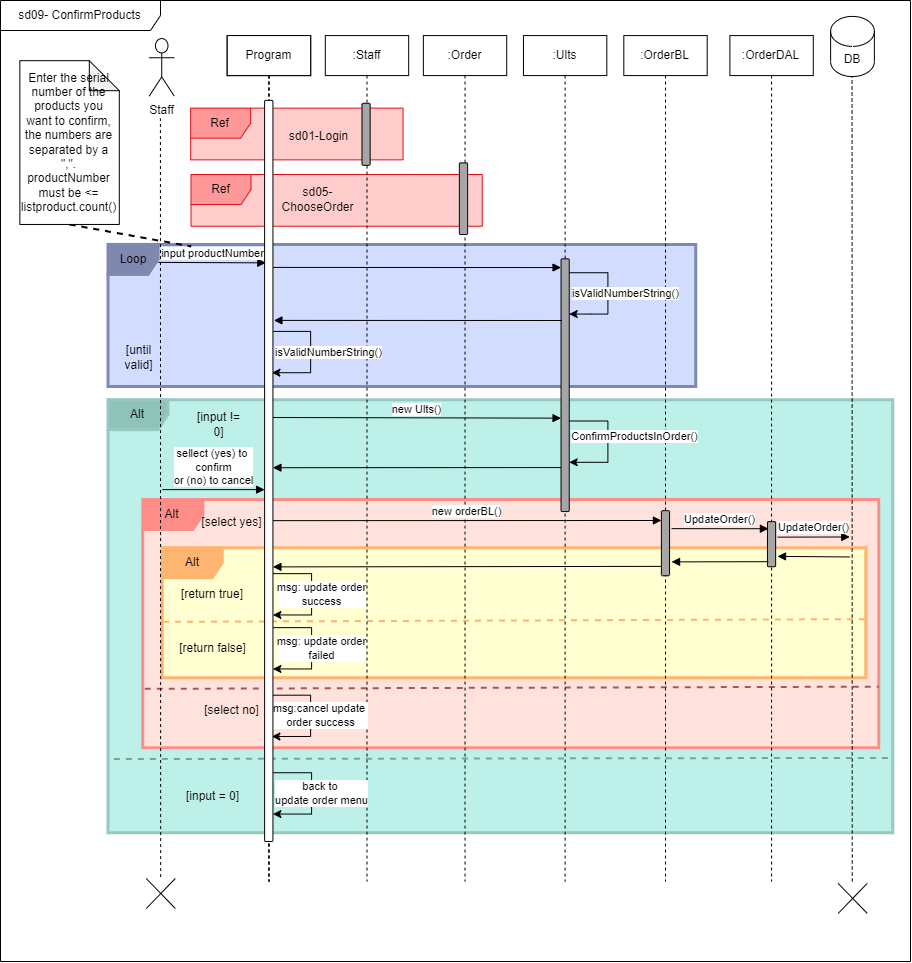
1. Code Design (Class Diagram)



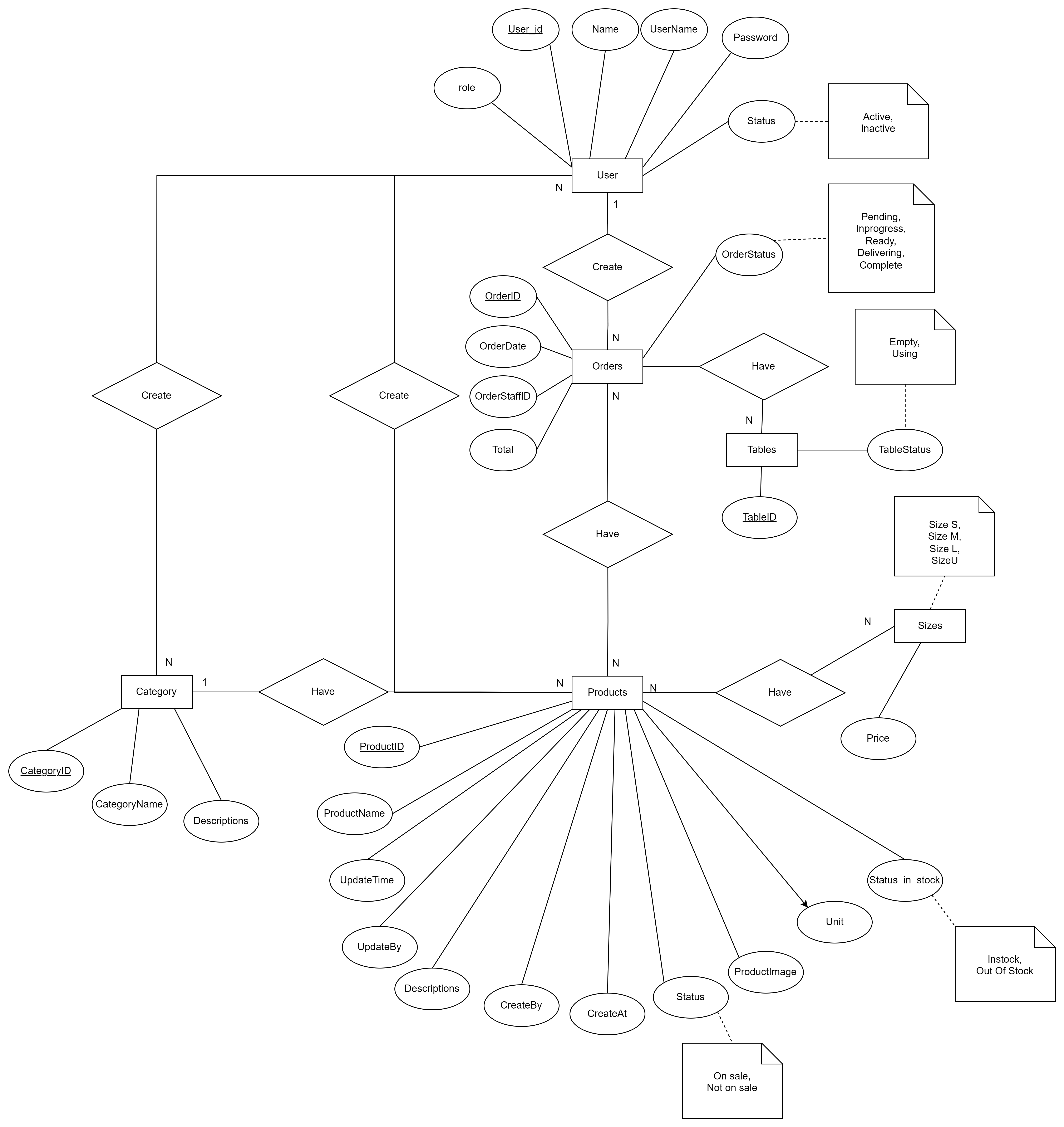
1. Sequence Diagram

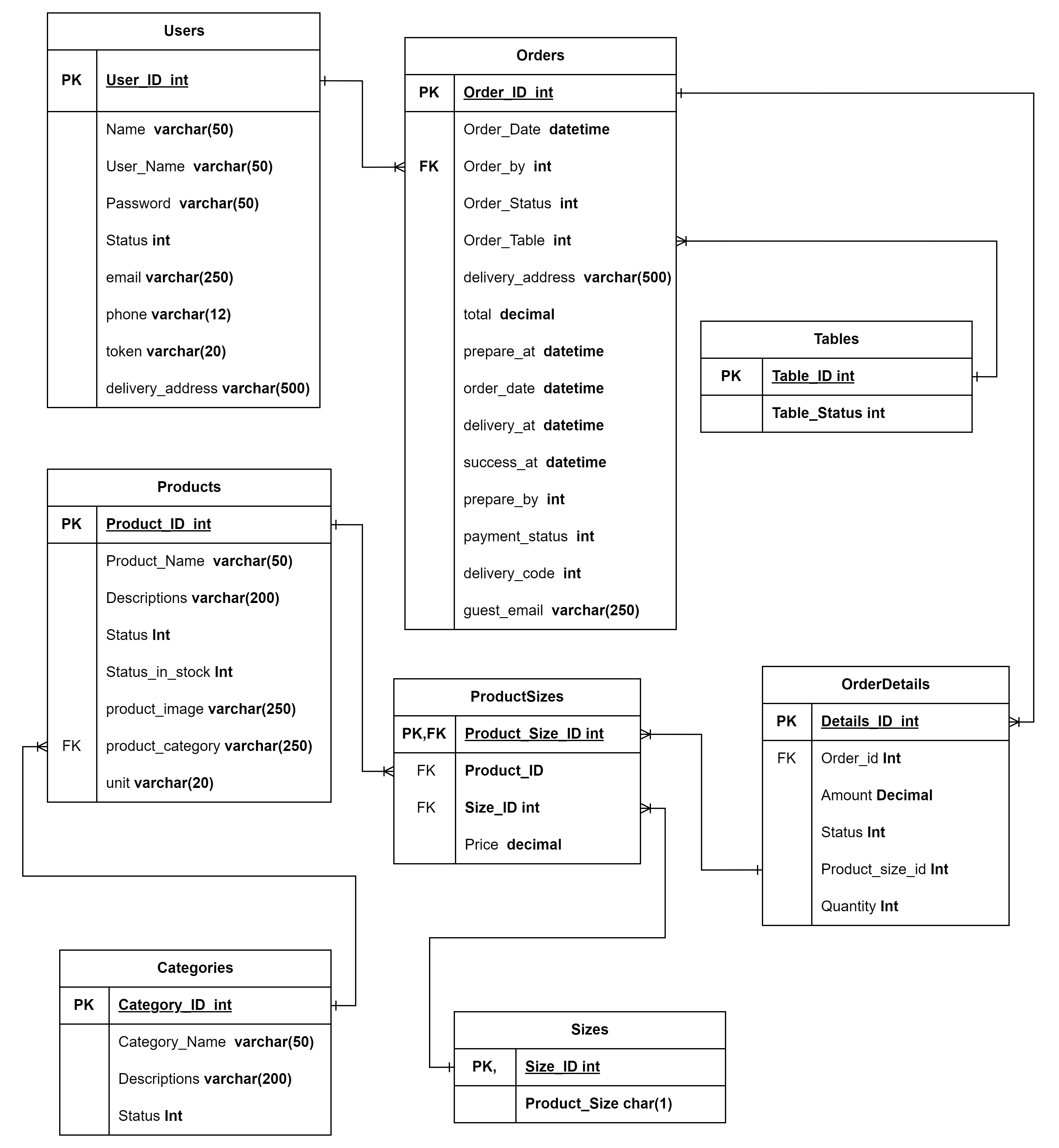
A screenshot of a computer screen

Description automatically generatedA screenshot of a computer

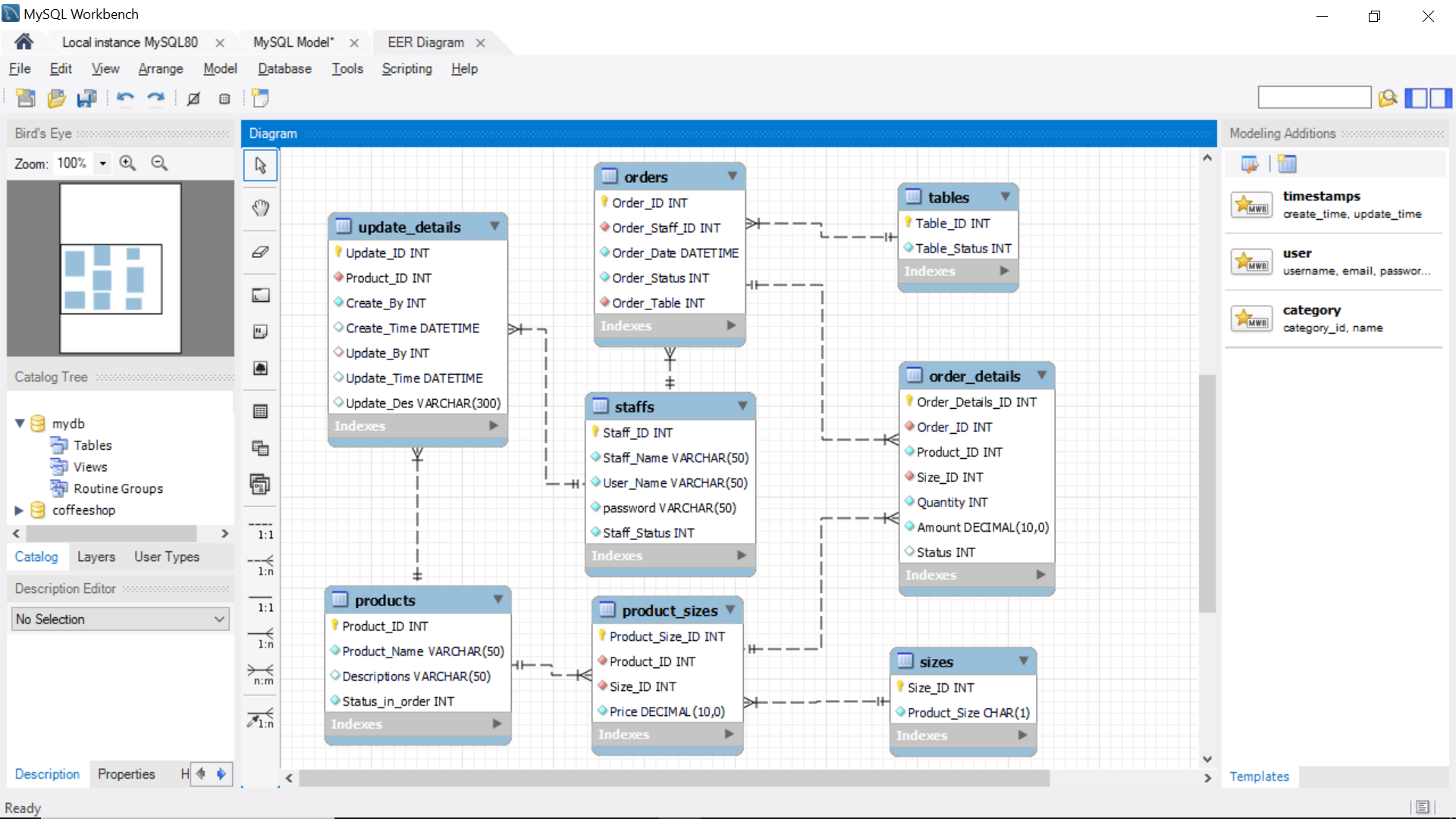
Description automatically generated

1. Database Design
   1. Entity Relationship Diagram





* 1. Database Design Details



|  |  |  |  |
| --- | --- | --- | --- |
| Staffs | | | |
| Column Name | Data Type | Constraints | Description |
| Staff\_ID | int | Primary Key, Auto Increment |  |
| Staff\_Name | varchar(50) | NOT NULL |  |
| User\_Name | varchar(50) | NOT NULL |  |
| password | varchar(50) | NOT NULL |  |
| Staff\_Status | int | NOT NULL |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Orders | | | |
| Column Name | Data Type | Constraints | Description |
| Order\_ID | int | Primary Key, Auto Increment |  |
| Order\_Staff\_ID | int | Foreign Key, NOT NULL |  |
| Order\_Date | DateTime | NOT NULL | Current Time |
| Order\_Status | Int | NOT NULL | 1. Inprogress  2. Complete |
| Order\_Table | int | Foreign Key, NOT NULL |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Products | | | |
| Column Name | Data Type | Constraints | Description |
| Product\_ID | int | Primary Key, Auto Increment |  |
| Product\_Name | varchar(50) | Foreign Key, NOT NULL |  |
| Description | varchar(50) |  | Product’s description |
| Status\_in\_order | int | NOT NULL | 1. Inprogress  2. Complete |

|  |  |  |  |
| --- | --- | --- | --- |
| Product\_sizes | | | |
| Column Name | Data Type | Constraints | Description |
| Product\_Size\_ID | int | Primary Key, Auto Increment |  |
| Product\_ID | int | Foreign Key, NOT NULL |  |
| Size\_ID | Int | Foreign Key, NOT NULL | 1. S 2. M 3. L |
| Price | Decimal(10,0) | NOT NULL | Product’s price |

|  |  |  |  |
| --- | --- | --- | --- |
| Tables | | | |
| Column Name | Data Type | Constraints | Description |
| Table\_ID | int | Primary Key, NOT NULL |  |
| Table\_Status | int | Foreign Key, NOT NULL | 1. Empty  2. Using |

|  |  |  |  |
| --- | --- | --- | --- |
| Order\_Details | | | |
| Column Name | Data Type | Constraints | Description |
| Order\_Details\_ID | int | Primary Key, Auto Increment |  |
| Order\_ID | int | Foreign Key, NOT NULL |  |
| Product\_ID | Int | NOT NULL |  |
| Size\_ID | int | Foreign Key, NOT NULL |  |
| Quantity | int | NOT NULL | Product’s quantity |
| Amount | Decimal(10,0) | NOT NULL | Amount product’s price |
| Status | Int | NOT NULL | 1. Inprogress  2. Complete |

|  |  |  |  |
| --- | --- | --- | --- |
| Update\_Details | | | |
| Column Name | Data Type | Constraints | Description |
| Update\_ID | int | Primary Key, NOT NULL |  |
| Product\_ID | int | Foreign Key, NOT NULL |  |
| Create\_By | Int | NOT NULL |  |
| Create\_Time | DateTime | NOT NULL | Order Creation Time |
| Update\_By | int | NOT NULL |  |
| Update\_Time | DateTime | NOT NULL | Current Time |
| Update\_DES | Varchar(300) | NOT NULL | Update Order Description |

|  |  |  |  |
| --- | --- | --- | --- |
| Sizes | | | |
| Column Name | Data Type | Constraints | Description |
| Size\_ID | int | Primary Key, NOT NULL |  |
| Product\_Size | char(1) | NOT NULL | 1. S 2. M 3. L |

# Test

1. Login test
   1. Login test 1

|  |  |
| --- | --- |
| Test Case Number | 001 |
| Test Case Name | LoginTest1() |
| Test Case Description | This test case checks username and password correct or not |
| Preconditions | Username and password must exist in the database |
| Test Case Input | “staff001”,”123456”  “1”,”1” |
| Test Case Expected Output | A Staff |
| Test Case Steps | New Staff  New StaffDAL  Call CheckAuthorize method  Compare output with expected output |
| Default Value Preverving |  |

* 1. Login test 2

|  |  |
| --- | --- |
| Test Case Number | 002 |
| Test Case Name | LoginTest2() |
| Test Case Description | This test case checks username and password correct or not |
| Preconditions | Username and password must exist in the database |
| Test Case Input | “staff001fhj”,”123456”  “staff001”,”277sj2”  “Staff01”,”12345da” |
| Test Case Expected Output | Null |
| Test Case Steps | New Staff  New StaffDAL  Call CheckAuthorize method  Compare output with expected output |
| Default Value Preverving |  |

1. Create order test
   1. Create order test 1

|  |  |
| --- | --- |
| Test Case Number | 001 |
| Test Case Name | CreateOrderTest1() |
| Test Case Description | Create an order with at least one product in menu |
| Preconditions | Logged in |
| Test Case Input | 1,10,2  10,3,4  6,1 |
| Test Case Expected Output | True |
| Test Case Steps | New Order  New OrderDAL  New ProductDAL  New TableDAL  Call GetProductById method  Call GetTableById method  Call CreateOrder method  Compare output with expected output |
| Default Value Preverving |  |

* 1. Create order test 2

|  |  |
| --- | --- |
| Test Case Number | 002 |
| Test Case Name | CreateOrderTest2() |
| Test Case Description | Create an order with at least one product in menu |
| Preconditions | Logged in |
| Test Case Input | 5,0  2,0 |
| Test Case Expected Output | Back to main menu. |
| Test Case Steps | New Order  New OrderDAL  New TableDAL  New ProductDAL  Call GetTableById method  Call GetProductById method  Compare output with expected output |
| Default Value Preverving |  |

* 1. Create order test 3

|  |  |
| --- | --- |
| Test Case Number | 003 |
| Test Case Name | CreateOrderTest3() |
| Test Case Description | Create an order with at least one product in menu |
| Preconditions | Logged in |
| Test Case Input | 5,3,4,2 |
| Test Case Expected Output | True |
| Test Case Steps | New Order  New OrderDAL  New ProductDAL  New TableDAL  Call GetProductById method  Call GetTableById method  Call CreateOrder method  Compare output with expected output |
| Default Value Preverving |  |
|  |  |

* 1. Create order test 4

|  |  |
| --- | --- |
| Test Case Number | 004 |
| Test Case Name | CreateOrderTest4() |
| Test Case Description | Create an order with at least one product in menu |
| Preconditions | Logged in |
| Test Case Input | 1,2,a  2,ba |
| Test Case Expected Output | False |
| Test Case Steps | New Order  New OrderDAL  New TableDAL  New ProductDAL  Call GetTableById method  Call GetProductById method  Call CreateOrder method  Compare output with expected output |
| Default Value Preverving |  |
|  |  |

* 1. Create order test 5

|  |  |
| --- | --- |
| Test Case Number | 005 |
| Test Case Name | CreateOrderTest5() |
| Test Case Description | Create an order with at least one product in menu |
| Preconditions | Logged in |
| Test Case Input | 10,41  1,49 |
| Test Case Expected Output | False |
| Test Case Steps | New Order  New OrderDAL  New ProductDAL  New TableDAL  Call GetTableById method  Call GetProductById method  Call CreateOrder method  Compare output with expected output |
| Default Value Preverving |  |
|  |  |

1. Update Order test
   1. Add product to order test
      1. Add product to order test 1

|  |  |
| --- | --- |
| Test Case Number | 001 |
| Test Case Name | AddProductToOrderTest1() |
| Test Case Description | Add one or more products to a previously created order |
| Preconditions |  |
| Test Case Input | 1,40,1 |
| Test Case Expected Output | True |
| Test Case Steps | New Order  New OrderDAL  New ProductDAL  Call ChooseOrder method  Call GetProductById method  Call UpdateOrder method  Compare output with expected output |
| Default Value Preverving |  |
|  |  |

* + 1. Add product to order test 2

|  |  |
| --- | --- |
| Test Case Number | 002 |
| Test Case Name | AddProductToOrderTest2() |
| Test Case Description | Add one or more products to a previously created order |
| Preconditions |  |
| Test Case Input | 1,40,0  0 |
| Test Case Expected Output | Back to update menu |
| Test Case Steps | New Order  New OrderDAL  New ProductDAL  Call ChooseOrder method  Call GetProductById method  Call UpdateOrder method  Compare output with expected output |
| Default Value Preverving |  |
|  |  |

* + 1. Add product to order test 3

|  |  |
| --- | --- |
| Test Case Number | 003 |
| Test Case Name | AddProductToOrderTest3() |
| Test Case Description | Add one or more products to a previously created order |
| Preconditions |  |
| Test Case Input | 1,51  aa |
| Test Case Expected Output | False |
| Test Case Steps | New Order  New OrderDAL  New ProductDAL  Call ChooseOrder method  Call GetProductById method  Call UpdateOrder method  Compare output with expected output |
| Default Value Preverving |  |
|  |  |

* 1. Remove unfinished products test
     1. Remove unfinished products test 1

|  |  |
| --- | --- |
| Test Case Number | 001 |
| Test Case Name | RemoveUnfinishedProductTest1() |
| Test Case Description | Remove unfinished products from the order |
| Preconditions |  |
| Test Case Input | 1,1  2,1 |
| Test Case Expected Output | True |
| Test Case Steps | New Order  New OrderDAL  Call ChooseOrder method  Call RemoveProductInOrder method  Call UpdateOrder method  Compare output with expected output |
| Default Value Preverving |  |
|  |  |

* + 1. Remove unfinished products test 2

|  |  |
| --- | --- |
| Test Case Number | 002 |
| Test Case Name | RemoveUnfinishedProductTest2() |
| Test Case Description | Remove unfinished products from the order |
| Preconditions |  |
| Test Case Input | 1,1,2  2,1,2 |
| Test Case Expected Output | True |
| Test Case Steps | New Order  New OrderDAL  Call ChooseOrder method  Call RemoveProductInOrder method  Call RemoveOrder method  Compare output with expected output |
| Default Value Preverving |  |
|  |  |

* + 1. Remove unfinished products test 3

|  |  |
| --- | --- |
| Test Case Number | 003 |
| Test Case Name | RemoveUnfinishedProductTest3() |
| Test Case Description | Remove unfinished products from the order |
| Preconditions |  |
| Test Case Input | 0  2,0 |
| Test Case Expected Output | Back to update menu. |
| Test Case Steps | New Order  New OrderDAL  Call ChooseOrder method  Call RemoveProductInOrder method  Call RemoveOrder method  Compare output with expected output |
| Default Value Preverving |  |
|  |  |

* + 1. Remove unfinished products test 4

|  |  |
| --- | --- |
| Test Case Number | 004 |
| Test Case Name | RemoveUnfinishedProductTest4() |
| Test Case Description | Remove unfinished products from the order |
| Preconditions |  |
| Test Case Input | Aa  10,aa  1,”1,a” |
| Test Case Expected Output | False |
| Test Case Steps | New Order  New OrderDAL  Call ChooseOrder method  Call RemoveProductInOrder method  Call RemoveOrder method  Compare output with expected output |
| Default Value Preverving |  |
|  |  |

* 1. Change product in order test
     1. Change product in order test 1

|  |  |
| --- | --- |
| Test Case Number | 001 |
| Test Case Name | ChangeProductTest1() |
| Test Case Description | Change one or more products in order |
| Preconditions |  |
| Test Case Input | 1,1,5,2  1,2,2,1 |
| Test Case Expected Output | True |
| Test Case Steps | New Order  New OrderDAL  Call ChooseOrder method  Call GetProductToChange method  Call UpdateOrder method  Compare output with expected output |
| Default Value Preverving |  |
|  |  |

* + 1. Change product in order test 2

|  |  |
| --- | --- |
| Test Case Number | 002 |
| Test Case Name | ChangeProductTest2() |
| Test Case Description | Change one or more products in order |
| Preconditions |  |
| Test Case Input | 1,0  2,1,0 |
| Test Case Expected Output | Back to main menu |
| Test Case Steps | New Order  New OrderDAL  Call ChooseOrder method  Call GetProductToChange method  Call UpdateOrder method  Compare output with expected output |
| Default Value Preverving |  |
|  |  |

* + 1. Change product in order test 3

|  |  |
| --- | --- |
| Test Case Number | 003 |
| Test Case Name | ChangeProductTest3() |
| Test Case Description | Change one or more products in order |
| Preconditions |  |
| Test Case Input | 0a  11,1a2s |
| Test Case Expected Output | False |
| Test Case Steps | New Order  New OrderDAL  Call ChooseOrder method  Call GetProductToChange method  Call UpdateOrder method  Compare output with expected output |
| Default Value Preverving |  |
|  |  |

* 1. Confirm product in order test
     1. Confirm product in order test 1

|  |  |
| --- | --- |
| Test Case Number | 001 |
| Test Case Name | ConfirmProductTest1() |
| Test Case Description | Change status one or more products in order to confirm |
| Preconditions |  |
| Test Case Input | 2,1,2  5,1 |
| Test Case Expected Output | True |
| Test Case Steps | New Order  New OrderDAL  Call ChooseOrder method  Call ConfirmProductsInOrder method  Call UpdateOrder method  Compare output with expected output |
| Default Value Preverving |  |
|  |  |

* + 1. Confirm product in order test 2

|  |  |
| --- | --- |
| Test Case Number | 002 |
| Test Case Name | ConfirmProductTest2() |
| Test Case Description | Change status one or more products in order to confirm |
| Preconditions |  |
| Test Case Input | 2,0  11,0 |
| Test Case Expected Output | Back to update menu |
| Test Case Steps | New Order  New OrderDAL  Call ChooseOrder method  Call ConfirmProductsInOrder method  Call UpdateOrder method  Compare output with expected output |
| Default Value Preverving |  |
|  |  |

* + 1. Confirm product in order test 3

|  |  |
| --- | --- |
| Test Case Number | 003 |
| Test Case Name | ConfirmProductTest3() |
| Test Case Description | Change status one or more products in order to confirm |
| Preconditions |  |
| Test Case Input | 2aa  23,a1 |
| Test Case Expected Output | False |
| Test Case Steps | New Order  New OrderDAL  Call ChooseOrder method  Call ConfirmProductsInOrder method  Call UpdateOrder method  Compare output with expected output |
| Default Value Preverving |  |
|  |  |

* 1. Change table test
     1. Change table test 1

|  |  |
| --- | --- |
| Test Case Number | 001 |
| Test Case Name | ChangeTableTest1() |
| Test Case Description | Change the table in the order to use at the coffee shop |
| Preconditions |  |
| Test Case Input | 1,2  10,5 |
| Test Case Expected Output | True |
| Test Case Steps | New Table  New TableDAL  Call ChooseOrder method  Call GetTableById method  Call ChangeTableOrder method  Compare output with expected output |
| Default Value Preverving |  |
|  |  |

* + 1. Change table test 2

|  |  |
| --- | --- |
| Test Case Number | 002 |
| Test Case Name | ChangeTableTest2() |
| Test Case Description | Change the table in the order to use at the coffee shop |
| Preconditions |  |
| Test Case Input | 1,0  10,0 |
| Test Case Expected Output | Back to update menu |
| Test Case Steps | New Table  New TableDAL  Call ChooseOrder method  Call GetTableById method  Call ChangeTableOrder method  Compare output with expected output |
| Default Value Preverving |  |
|  |  |

* + 1. Change table test 3

|  |  |
| --- | --- |
| Test Case Number | 003 |
| Test Case Name | ChangeTableTest3() |
| Test Case Description | Change the table in the order to use at the coffee shop |
| Preconditions |  |
| Test Case Input | 1,1  10,1  1,aa |
| Test Case Expected Output | False |
| Test Case Steps | New Table  New TableDAL  Call ChooseOrder method  Call GetTableById method  Call ChangeTableOrder method  Compare output with expected output |
| Default Value Preverving |  |
|  |  |

1. Update product status in stock test
   1. Update product status in stock test 1

|  |  |
| --- | --- |
| Test Case Number | 001 |
| Test Case Name | UpdateProductStatusInStockTest1() |
| Test Case Description | Change product status |
| Preconditions | Logged in |
| Test Case Input | 1  20 |
| Test Case Expected Output | True |
| Test Case Steps | New Product  New ProductDAL  Call GetProductInstockById method  Call ChangeProductStatus method  Compare output with expected output |
| Default Value Preverving |  |
|  |  |

* 1. Update product status in stock test 2

|  |  |
| --- | --- |
| Test Case Number | 002 |
| Test Case Name | UpdateProductStatusInStockTest2() |
| Test Case Description | Change product status |
| Preconditions | Logged in |
| Test Case Input | 0 |
| Test Case Expected Output | Back to main menu |
| Test Case Steps | New Product  New ProductDAL  Call GetProductInstockById method  Call ChangeProductStatus method  Compare output with expected output |
| Default Value Preverving |  |
|  |  |

* 1. Update product status in stock test 3

|  |  |
| --- | --- |
| Test Case Number | 003 |
| Test Case Name | UpdateProductStatusInStockTest3() |
| Test Case Description | Change product status |
| Preconditions | Logged in |
| Test Case Input | 2  13 |
| Test Case Expected Output | Back to main menu. |
| Test Case Steps | New Product  New ProductDAL  Call GetProductInstockById method  Call ChangeProductStatus method  Compare output with expected output |
| Default Value Preverving |  |
|  |  |

* 1. Update product status in stock test 4

|  |  |
| --- | --- |
| Test Case Number | 004 |
| Test Case Name | UpdateProductStatusInStockTest4() |
| Test Case Description | Change product status |
| Preconditions | Logged in |
| Test Case Input | a  @1  100 |
| Test Case Expected Output | False |
| Test Case Steps | New Product  New ProductDAL  Call GetProductInstockById method  Call ChangeProductStatus method  Compare output with expected output |
| Default Value Preverving |  |
|  |  |

1. Payment Test

5.1. Payment Test 1

|  |  |
| --- | --- |
| Test Case Number | 001 |
| Test Case Name | PaymentTest1() |
| Test Case Description | Confirm payment |
| Preconditions | Logged in |
| Test Case Input | 2 |
| Test Case Expected Output | True |
| Test Case Steps | 1. Confirm product in order. 2. Select Payment in main menu. 3. Input order ID you want to confirm. 4. Confirm payment. |
| Default Value Preverving | New Order  New OrderDAL  Call ChooseOrder method  Call CompleteOrder method |
|  |  |

* 1. Payment Test 2

|  |  |
| --- | --- |
| Test Case Number | 002 |
| Test Case Name | PaymentTest2() |
| Test Case Description | Confirm payment |
| Preconditions | Logged in |
| Test Case Input | 5  1 |
| Test Case Expected Output | False |
| Test Case Steps | New Order  New OrderDAL  Call ChooseOrder method  Call CompleteOrder method |
| Default Value Preverving |  |
|  |  |

* 1. Payment Test 3

|  |  |
| --- | --- |
| Test Case Number | 003 |
| Test Case Name | PaymentTest3() |
| Test Case Description | Confirm payment |
| Preconditions | Logged in |
| Test Case Input | 0 |
| Test Case Expected Output | Back to main menu |
| Test Case Steps | New Order  New OrderDAL  Call ChooseOrder method  Call CompleteOrder method |
| Default Value Preverving |  |
|  |  |

1. Check Out Test
   1. Check Out Test 1

|  |  |
| --- | --- |
| Test Case Number | 001 |
| Test Case Name | CheckOutTest1() |
| Test Case Description | Can use to check out |
| Preconditions | Logged in |
| Test Case Input |  |
| Test Case Expected Output | True |
| Test Case Steps | New Staff  New StaffDAL  Call UpdateLogOutTime method |
| Default Value Preverving |  |

* 1. Check Out Test 2

|  |  |
| --- | --- |
| Test Case Number | 002 |
| Test Case Name | CheckOutTest2() |
| Test Case Description | Can use to check out |
| Preconditions | Logged in |
| Test Case Input | 0 |
| Test Case Expected Output | Back to main menu. |
| Test Case Steps | New Staff  New StaffDAL  Call UpdateLogOutTime method |
| Default Value Preverving |  |
|  |  |

# Task Assign (to each team member)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Group Name | Project Name | | | | | |
| No | Task name | Description | Start Date | End Date | Member | Self assessment |
| 1 | Write report |  | June 8 | Aug 31 | Duc & Ly |  |
| 2 | Draw diagram |  | June 8 | Aug 31 | Duc & Ly |  |
| 3 | Code |  | June 9 | Aug 31 | Duc & Ly |  |
| 4 | Write Slide |  | Aug 15 | Aug 31 | Duc & Ly |  |
| 5 |  |  |  |  |  |  |

# Installation Instructions

**INSTALLATION INSTRUCTIONS FOR CSM APPLICATION**

System Requirements :

* Operating System : Windows
* Version (.Net) : .Net7.0

1. Deployment Diagram

A black and white image of a rectangular object

Description automatically generated

1. Installation steps
   * Database Install

* Download MySQL Community Sever in website <https://dev.mysql.com/downloads/mysql/>

A screenshot of a computer

Description automatically generated

* Choose the version suitable for your device and operating system and press Dowload
* Extract the file and perform the installation process.
* Click "Next" in turn to navigate through the installation steps. You can change the installation options like installation directory, port, root password, etc.
* When the installation is complete, the installer will provide the option to configure MySQL and start the MySQL service.
* Open Command Prompt and run the command: mysql --version to check the installed MySQL version.
  + Application install
* Click on this link <https://drive.google.com/drive/folders/1PMa9kj4YhENkSs6bi1ngei0lKr4_HiRH?usp=drive_link>

and download the CSMApp\_Publish\_Version\_1.0.0 folder.

* Once downloaded, double-click the CSMApp\_Publish\_Version\_1.0.0 folder and copy the folder's path
* Open Command Port and use the following command:

cd <paste the copied folder path>

mysql -u root -p < DB.sql

then enter your password of root account in MySQL

* Go back to the CSMApp\_Publish\_Version\_1.0.0 folder, right click on CSMApp.exe and select Send to > Desktop(create shortcut) then go to desktop and use application