VIETNAM GENERAL CONFEDERATION OF LABOR

**TON DUC THANG UNIVERSITY**

**FACULTY OF INFORMATION TECHNOLOGY**



**FINAL PROJECT**

**SOFTWARE ENGINEERING**

*Student*: **TRẦN THỊ MỸ NHI – 518H0543**

**VI THỊ NGỌC YẾN – 519H0264**

Class **:**

Year  **: 22**

*Lecturer*:  **Dr. PHẠM THÁI KỲ TRUNG**

**HO CHI MINH CITY, 2022**

THANK YOU

This is a short self-written note expressing my gratitude to those who have helped me complete the Project. Thanks to the guidance from Mr. Phạm Thái Kỳ Trung, a lecturer in this subject, we were able to complete this project. Thank you very much sir.

**PROJECT COMPLETED**

**AT TON DUC THANG UNIVERSITY**

I hereby declare that this is our own project and is under the guidance of Phạm Thái Kỳ Trung. The research contents and results in this topic are honest and have not been published in any publication before. The data in the tables for analysis, comments and evaluation are collected by the author himself from different sources, clearly stated in the reference section.

In addition, the project also uses a number of comments, assessments as well as data of other authors, other agencies and organizations, with citations and source annotations.

If I find any fraud, I will take full responsibility for the content of my project. Ton Duc Thang University is not related to copyright and copyright violations caused by me (if any).

*Ho Chi Minh, date*

*Author*

*(Sign)*

*Trần Thị Mỹ Nhi*

*Vi Thị Ngọc Yến*

LECTURER’S ASSESSMENT

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*Ho Chi Minh, date*

*(sign)*

CONTENT

[THANK YOU i](#_bookmark0)

[PROJECT COMPLETEED AT TON DUC THANG UNIVERSITY 1](#_bookmark1)

[LECTURE’S ASSESSMENT 1](#_bookmark2)

[CHAPTER I – INTRODUCTION 3](#_bookmark3)

* 1. [Purpose and Scope 3](#_bookmark4)
  2. [Product Overview 3](#_bookmark5)

[CHAPTER II – PROJECT MANAGEMENT PLAN 5](#_bookmark6)

* 1. [Project Organization 5](#_bookmark7)
  2. [Lifecycle Model Used 5](#_bookmark8)

[CHAPTER III – REQUIREMENT SPECIFICATIONS 7](#_bookmark9)

* 1. [Stakeholders for the system 7](#_bookmark10)
  2. [Use case model 8](#_bookmark11)
  3. [Functional requirements 18](#_bookmark12)
  4. [Non-functional requirements 18](#_bookmark13)

[CHAPTER IV – DESIGN 20](#_bookmark19)

* 1. [Database design 20](#_bookmark20)
  2. [Static model – class diagrams 23](#_bookmark21)
  3. [Dynamic model – sequence diagrams 24](#_bookmark22)

[CHAPTER V – TEST PLAN 26](#_bookmark23)

* 1. [Requirements/specifications-based system level test cases 26](#_bookmark24)
  2. [Unit test 27](#_bookmark25)
  3. [GIT/SVN 33](#_bookmark26)

[REFERENCES 34](#_bookmark27)

# CHAPTER I – INTRODUCTION

## Purpose and Scope

This software to manage a company selling functional foods. With functions such as creating warehouse receipts of many items, placing order lists, choosing payment methods, viewing order status, updating order status or reviewing monthly statistics. It is like a website selling goods for consumers and customers and employees all have functions for their own use on the website.

## Product Overview

* + 1. **Scenario for Accountant to Create Received Goods Note:**
* When a Company wants to receive more goods, an Accountant will interact with a software in order to create a Received Goods Note.
* This action contains Add product/ Remove product/ Update product quantity
* The result of this action is that the software will return a Receive Goods Note which has full information such as received note id, time, products’ details, price, …
  + 1. **Scenario for Accountant to Create Delivery Goods Note:**
* When a Company wants to deliver goods to agents, an Accountant will interact with a software in order to create a Delivery Goods Note.
* This action contains Add product/ Remove product/ Update product quantity/ Update payment status/ Update delivery status
* The result of this action is that the software will return a Delivery Goods Note which has full information such as delivery note id, time, products’ details, price, agent’s detail, payment status, delivery status, …
  + 1. **Scenario for Accountant to View Incoming/Outgoing Stock Report:**
* When an Accountant wants to look for a Stock Report of a randomly selected month, he/ she will interact with a software in order to View Report.
* This action contains View Incoming Stock/ View Outcoming Stock
* The result of this action is that the software will return a list of Incoming Products, or a list of Outcoming Products of that picked month.
  + 1. **Scenario for Accountant to View Revenue Report:**
* When an Accountant wants to look for a Revenue Report of a randomly selected month, he/ she will interact with a software in order to View Report.
* The result of this action is that the software will return a Revenue Report list of that picked month.
  + 1. **Scenario for Agents to Place An Order:**
* When Agents want to place an order, they will go to the Company website.
* This action allows Agents to add product to cart / update product quantity.
* Then the Agents check out their cart and click Choose Payment Method.
* The Agents will pay for the order and their order will be prepared.

- Then when you want to see the status of the order, the agent can go to the website to see the status of the order.

# CHAPTER II – PROJECT MANAGEMENT PLAN

## Project Organization

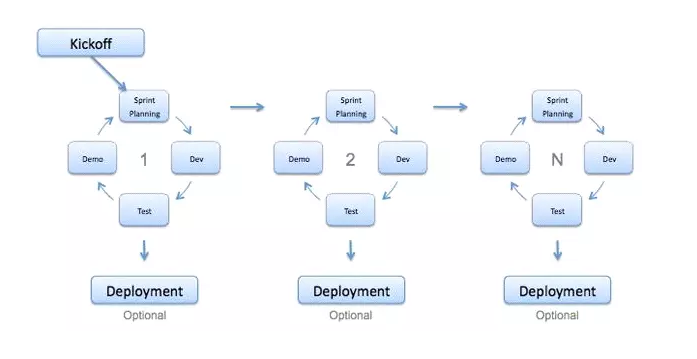
Divided into two directions: One is the accountant and the other is the Agent/Customer.

* Accountant includes Goods received note, Create goods delivery note, Update order status, Check statistic record, Check revenue record.
* Agent/Customer has Register, Login, Place order.

In each function, there are other items such as Payment, which have ID paymentDate, order ID, ...

## Lifecycle Model Used

In this project we decided to use Agile Model. Based on iterative and incremental model The requirements and solutions develop based on the combination of functions. It can be used with any type of project, but it requires customer engagement and interactivity. In addition, it can be used when customers require functionality to be available within a short period of time.



Advantages:

* Reduce the time it takes to take advantage of some system features
* The end result is high quality software in the least possible time and customer satisfaction

Defect:

* Depends on the skill of the software developer
* Documentation is done at the following stage
* Need an experienced team

In the Agile model we use the Scrum process which is a framework in which people can identify complex adaptive problems, while remaining productive and creative to deliver effective products. highest value.

* 1. **Risk Analysis**

Agile risk management is quite simple: eliminate risk

* Integrating early and often helps to reduce the risk that late integration of code poses.
* Also, early and frequent testing of the software minimizes the risk of finding bugs late when things are already in place making it difficult to test and change.
* And delivering early and often to customers reduces the biggest risk: Products no longer match customer needs.

In short, Agile Software Production helps to increase information and reduce risk in the first place. With Waterfall, we don't get much information until it's too late. At that point change is difficult, expensive and carries more risk.



Early and frequent integration, testing, and provisioning means constant and constant feedback. This reduces the risk of the project significantly.

* 1. **Hardware and Software Resource Requirements**
* **Software:**

An ERP system should have the following characteristics: Modular design includes many separate business modules such as finance, manufacturing, accounting, distribution, etc. Using a centralized common database management system (DBMS) Modules are integrated and provide seamless data flow between modules, increasing operational transparency through standard interfaces They are often complex systems with high costs They are flexible and offer best business practices They require time-consuming configuration and adjustment to integrate with the company's business functions real-time modules with online and batch processing capabilities They are or will soon support the Internet Various ERP vendors offer ERP systems with some degree of specialization but different models. The core module is almost the same for all of them. Some of the core ERP modules found in successful ERP systems are as follows: Accounting Management Financial Management Production Management Production Management Shipping Management Sales

* 1. **Deliverables and Schedule**
* Sprint 1 is to create a goods received note with a list of products.
* Sprint 2 is to create an order agent page with a view of the purchase schedule. Includes view of order details.
* Sprint 3 is to make a list of order lists, when you click on it, it will show details. Create a deli note to export to pdf and display the list of created applications.

# CHAPTER II – REQUIREMENT SPECIFICATONS

**3.1 Stakeholders for the system**

In this project, we noticed that there are 3 Stakeholders for the system, include Accountant , Agent and Customer:

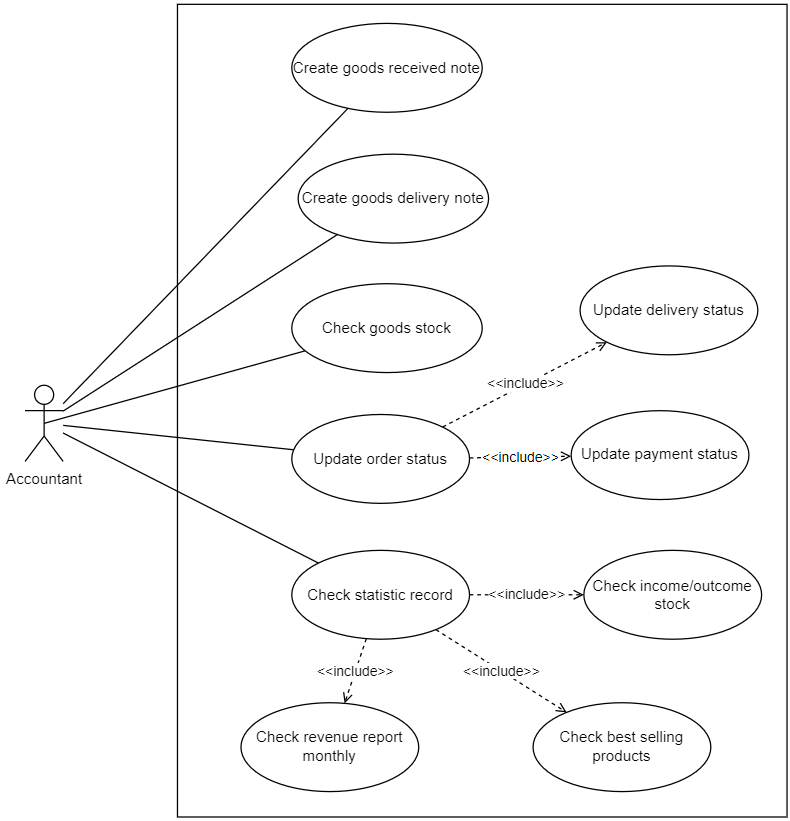
- Accountant: Who can create Goods Received Note and Goods Delivery Note, view Inventory Reports and View Statistic Reports, update order status on Window Form Application.

- Agent: Who can order goods from our Company through the Website with discount, choose payment method and check for order status.

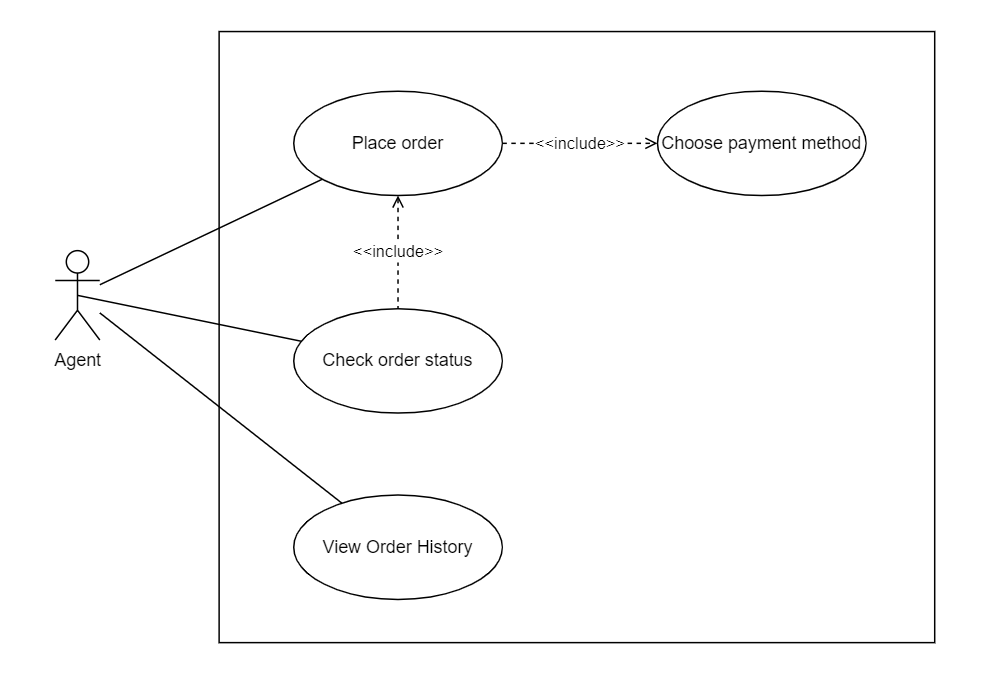
- Customer: Who can place order from our Company through the Website, choose payment method and check for order status.

**3.2. Use case model**

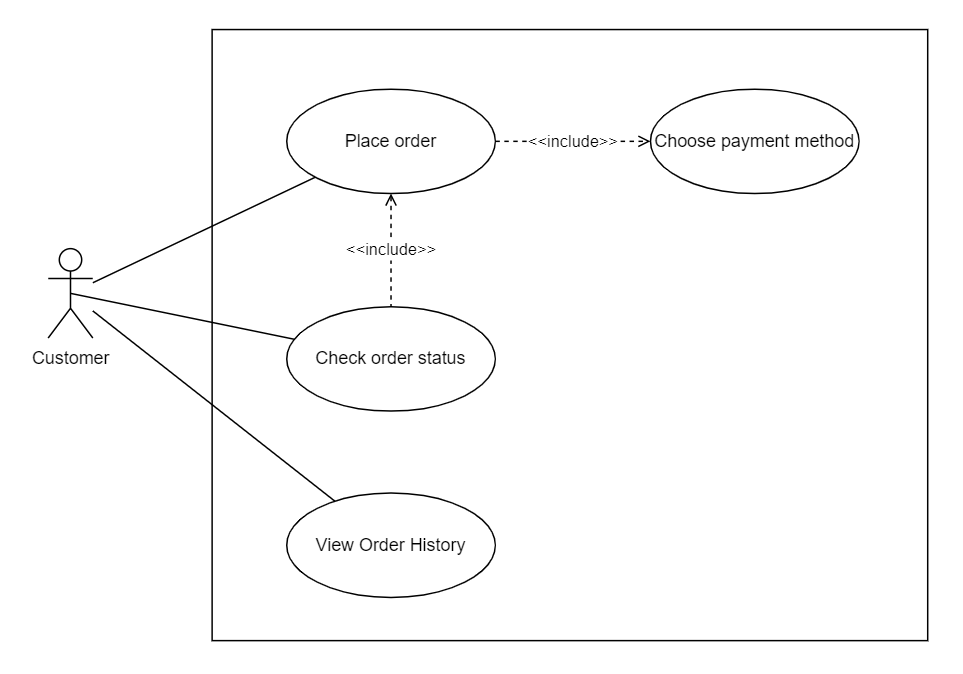
**3.2.1. Graphical use case model**

****

*Usecase Diagram for Accountant*



*Usecase Diagram for Agent*



**3.2.2.Textual Description for each use case**

* **Use case Check Goods Stock**

|  |  |
| --- | --- |
| **ID:** | **checkGoodsStock** |
| **Title:** | **Check Goods Stock** |
| **Description:** | * When an Accountant wants to look for a list of goods stock, she will interact with a software in order to View List. * This action contains View Goods List. * The result of this action is that the software will return a list of Goods include stock. |
| **Primary Actor:** | Accountant |
| **Preconditions:** | Must have goods item first. |
| **Postconditions:** | [Describe the state the system is in after all the events in this  use case have taken place.] |
| **Main Success Scenario:** | Accountant clicks on Goods List tab  Then the list of goods will be displayed |
| **Extensions:** | [Describe all the other scenarios for this use case - including  exceptions and error cases.] |
| **Frequency of Use:** | Frequently |
| **Status:** | [Development status] |
| **Owner:** | Tran Thi My Nhi |
| **Priority:** | [Priority of this use case] |

* **Use case Create Goods Receive Note**

|  |  |
| --- | --- |
| **ID:** | **createGoodsReceiveNote** |
| **Title:** | **Create Goods Receive Note** |
| **Description:** | * When the Company wants to receive more goods, the Accountant will interact with the software to make the goods receipt. * This action includes Add product/ Removeproduct/ Update product quantity. * The result of this operation is that the software will return a Goods Receive Note with informations such as goods ID , note create date, import price, quantity, total. |
| **Primary Actor:** | Accountant |
| **Preconditions:** | Product list and quantity must be available first. |
| **Postconditions:** | Goods stock is updated to the database. |
| **Main Success Scenario:** | Accountant clicks on Goods Receive Note tab   * Choose goods name * Enter quantity * Click on Add to add recently choosen product * Click on Remove to remove the product * Click on Create to create Goods Receive Note and save record. |
| **Extensions:** | If Accountant doesn’t add any product, accountant cannot click on Create button to Create Goods Received Note. |
| **Frequency of Use:** | Frequently |
| **Status:** | [Development status] |
| **Owner:** | Tran Thi My Nhi |
| **Priority:** | [Priority of this use case] |

* **Use case Create Goods Delivery Note**

|  |  |
| --- | --- |
| **ID:** | **createGoodsDeliveryNote** |
| **Title:** | **Create Goods Delivery Note** |
| **Description:** | * When the Company wants to export goods to the agent/, the Accountant will interact with the software to make the goods release note. * The result of this operation is that the software will return a warehouse release note with full information such as delivery note code, order creation date, delivery address, product ID, product name, price, quality quantity, payment status, quantity |
| **Primary Actor:** | Accountant |
| **Preconditions:** | List of products must be available before |
| **Postconditions:** | Outcoming products’ details will be added in database in  order for Accountant to view stock report |
| **Main Success Scenario:** | - Accountant clicks on Create Goods Delivery Note tabs  - Choose an order to view its detail products  - Click on Delivery/Print to create a delivery note |
| **Extensions:** | If Accountant doesn’t add any product, accountant cannot click on Receive Button to Create Goods Received Note. |
| **Frequency of Use:** | Frequently |
| **Status:** | [Development status] |
| **Owner:** | Tran Thi My Nhi |
| **Priority:** | [Priority of this use case] |

* **Use case View Statistic**

|  |  |
| --- | --- |
| **ID:** | **viewStatistic** |
| **Title:** | **View Statistic** |
| **Description:** | * When the Accountant wants to look up the Revenue Report of the selected month and will interact with the software to view the Report. * The result of this operation is that the software will return a list of sales report and a list of receipt report of the selected month. |
| **Primary Actor:** | Accountant |
| **Preconditions:** | The list of Sales Report of the selected month must exist |
| **Postconditions:** | The full information of revenue for the picked month will be  presented on the screen |
| **Main Success Scenario:** | Accountant clicks Statistic tab  Choose month  Choose year  Click Go button |
| **Extensions:** | None |
| **Frequency of Use:** | Not frequently, just few times every month |
| **Status:** | [Development status] |
| **Owner:** | Tran Thi My Nhi |
| **Priority:** | [Priority of this use case] |

* **Use case Place Order**

|  |  |
| --- | --- |
| **ID:** | **placeOrder** |
| **Title:** | **Place Order** |
| **Description:** | - When the agent/customer wants to buy a goods, they will go to the company's website, enter personal information, find the goods that they are looking for, and select the quantity. Then there will show total price and they will click on the submit button. This process will end after success |
| **Primary Actor:** | Agent / Customer |
| **Preconditions:** | The list of products in stock must be available first |
| **Postconditions:** | The place order information will be sent to the database for  the accountant to create a delivery goods note |
| **Main Success Scenario:** | Agent/customer goes to the company website  Choose the products that he/she is looking for and fill its quantity  Choose payment method  The process will finish after the payment is confirmed |
| **Extensions:** | If the payment is declined, the process cannot be done |
| **Frequency of Use:** | Frequently |
| **Status:** | [Development status] |
| **Owner:** | Tran Thi My Nhi |
| **Priority:** | [Priority of this use case] |

**3.3. Functional requirements**

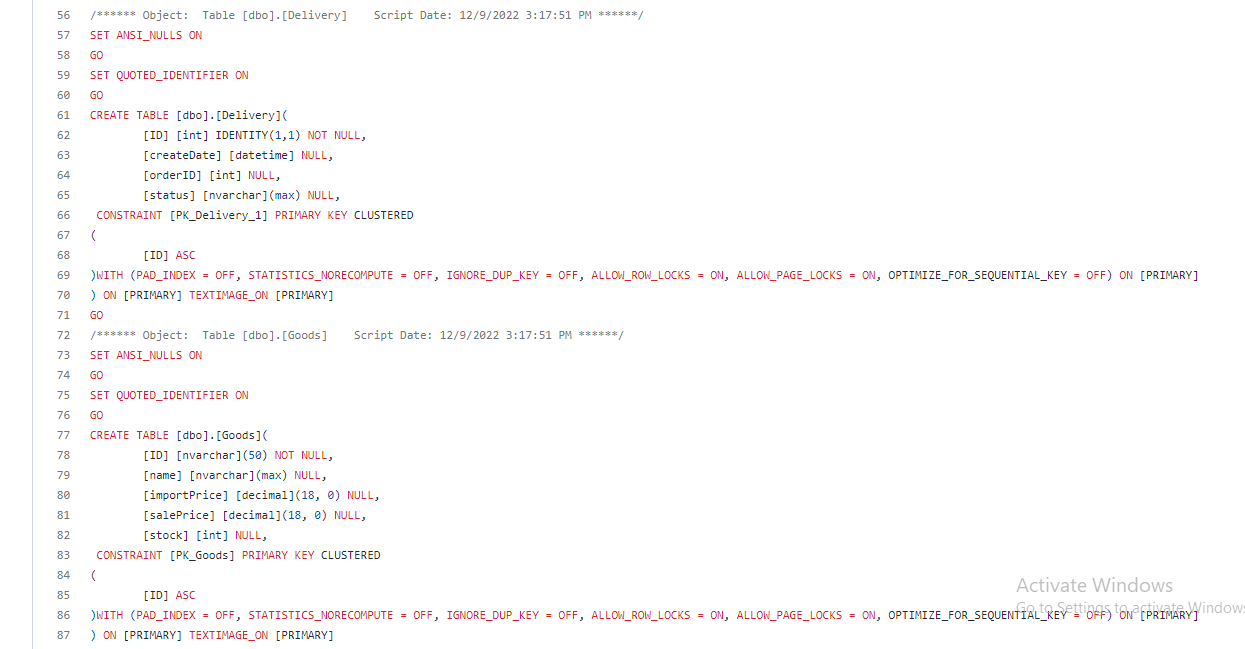
* ComboBox of goods name must contains goods name
* When creating tab Get Notes there must be a textBox to fill in the number of products
* Must have ListView to display data of Notes Get product details, Orders, Order details, Sales reports, Inventory reports
* There must be a final price for both the Receipt/Delivery Note
* On the Report tab must contain a comboBox to select the month, a textBox to enter the year and a comboBox to select
* To see the Export/Receive report - Full product catalog is required
* There must be a textBox for Dealers/customers to fill in the number of products
* Must have different payment method options (MOMO, cash, bank transfer)
* Must have a textBox for Dealers/Customers to fill in the number of products
* Must have different payment method options (MOMO, cash, bank transfer)

## 3.4 Non-functional requirements

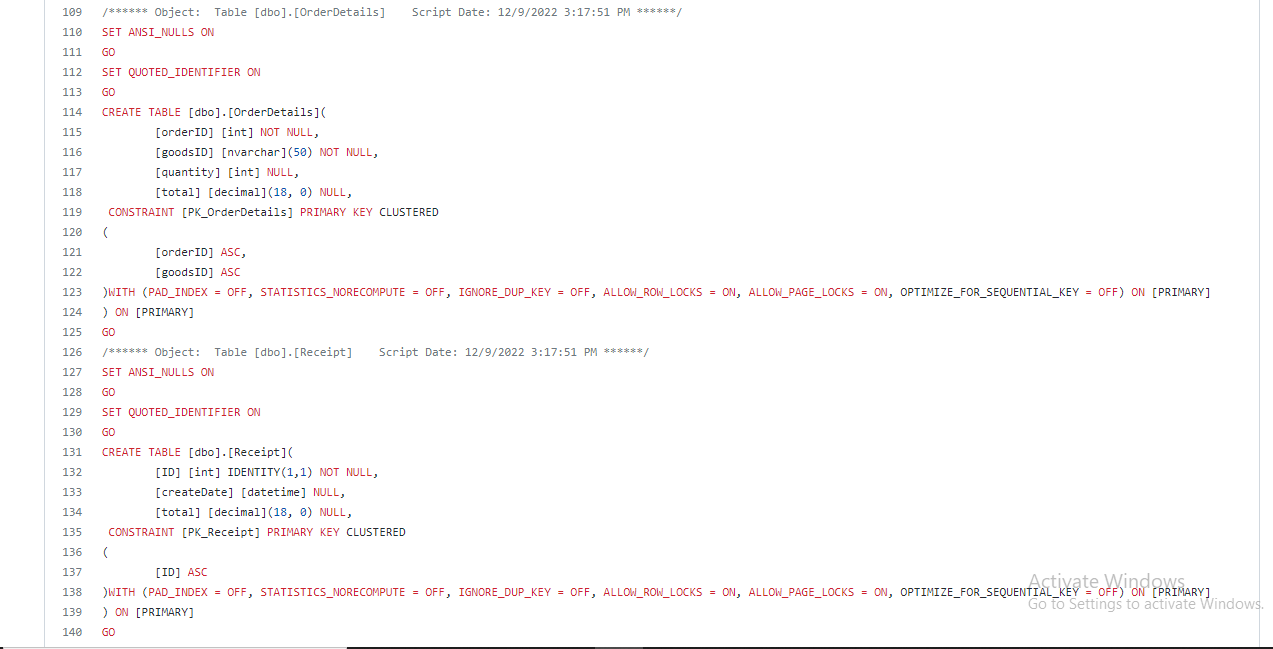
* Notes of goods received must be updated as soon as the Accountant presses the button to create notes
* The delivery note must be updated as soon as the accountant presses the button to create the voucher
* Warehouse receipt/dispatch report must be
* presented immediately after the Accountant made
* Statistic Reports will be presented correctly and right immediately after the Accountant clicks on Go button
* Time to execute payment should not be delayed more than 3 minutes

# CHAPTER IV – DESIGN

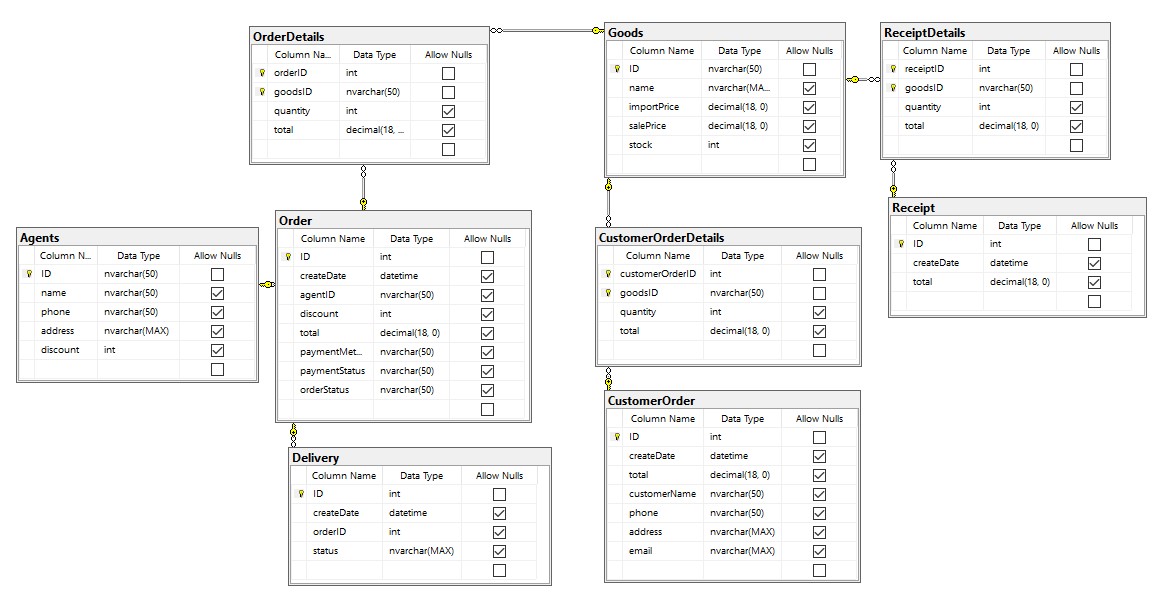
**4.1 Database design**



*A code to generate database*



*A code to generate database*

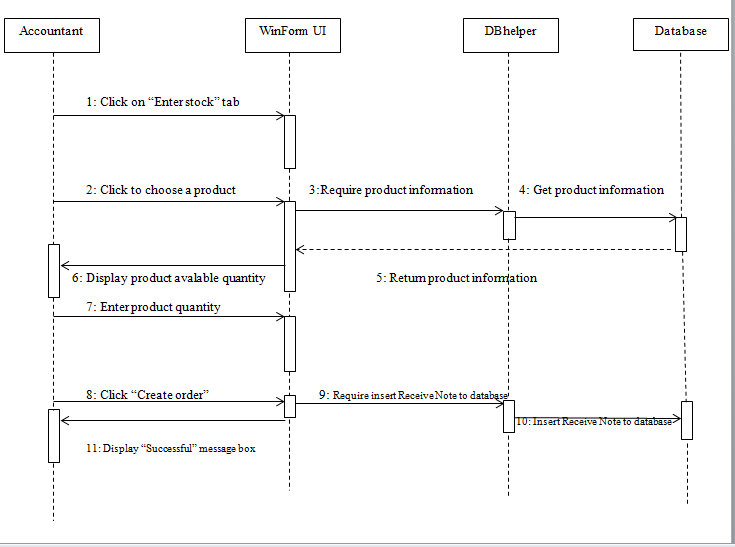


*Database Diagram*

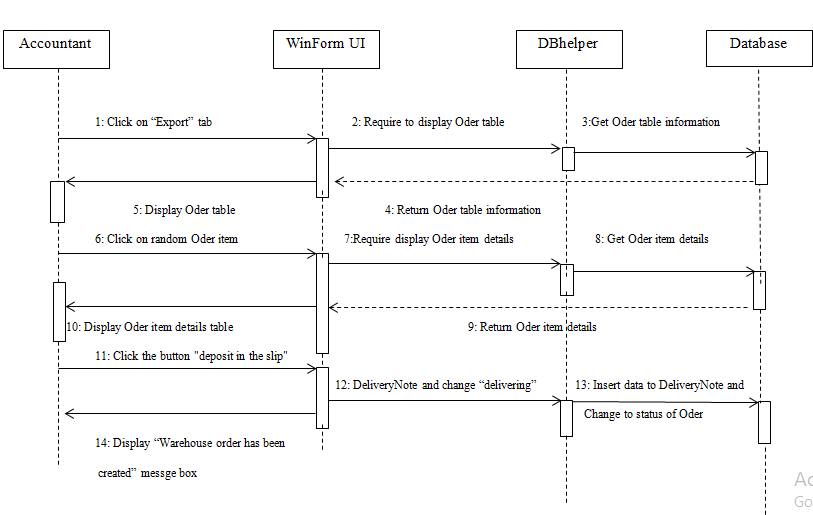
**5.2 Static model – class diagrams**

## F:\TRYHARD\HK1 - 20222023\CNPM\SE-Final-Project NOT FOR SUBMIT\images\ClassDiagram1.png

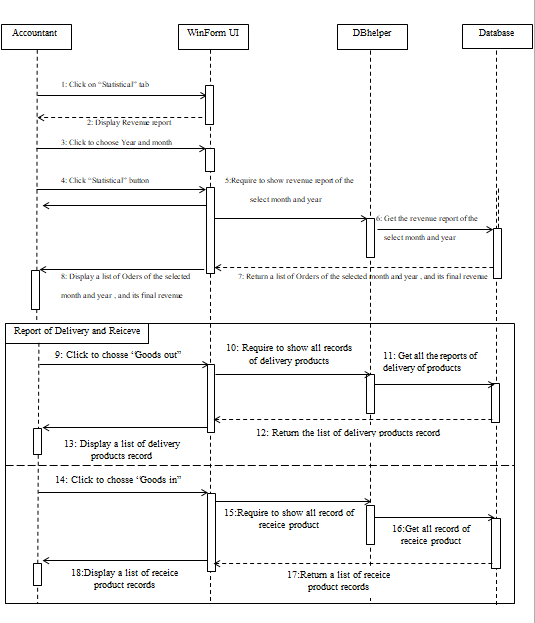
## 5.3 Dynamic model – sequence diagrams

****

*Sequence Diagram – Create Goods Receive Note*

****

*Sequence Diagram – Create Goods Delivery Note*

****

*Sequence Diagram – Create Goods Receive Note*

# CHAPTER VI – TEST PLAN

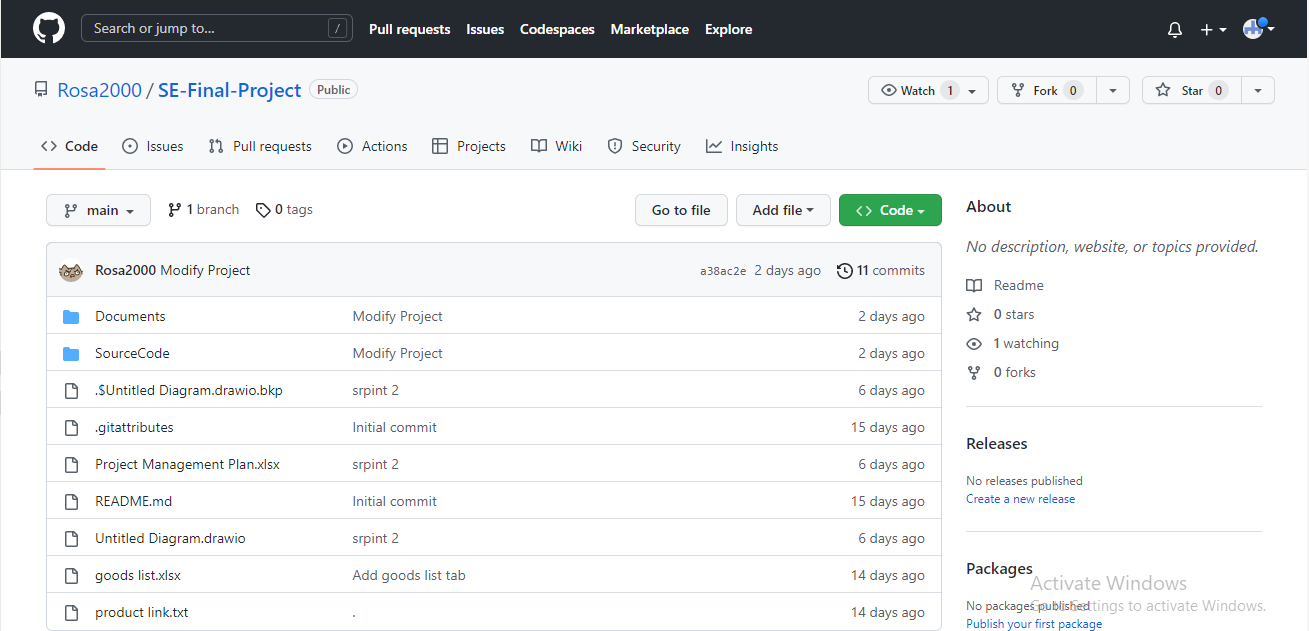
## Requirements/specifications-based system level test cases

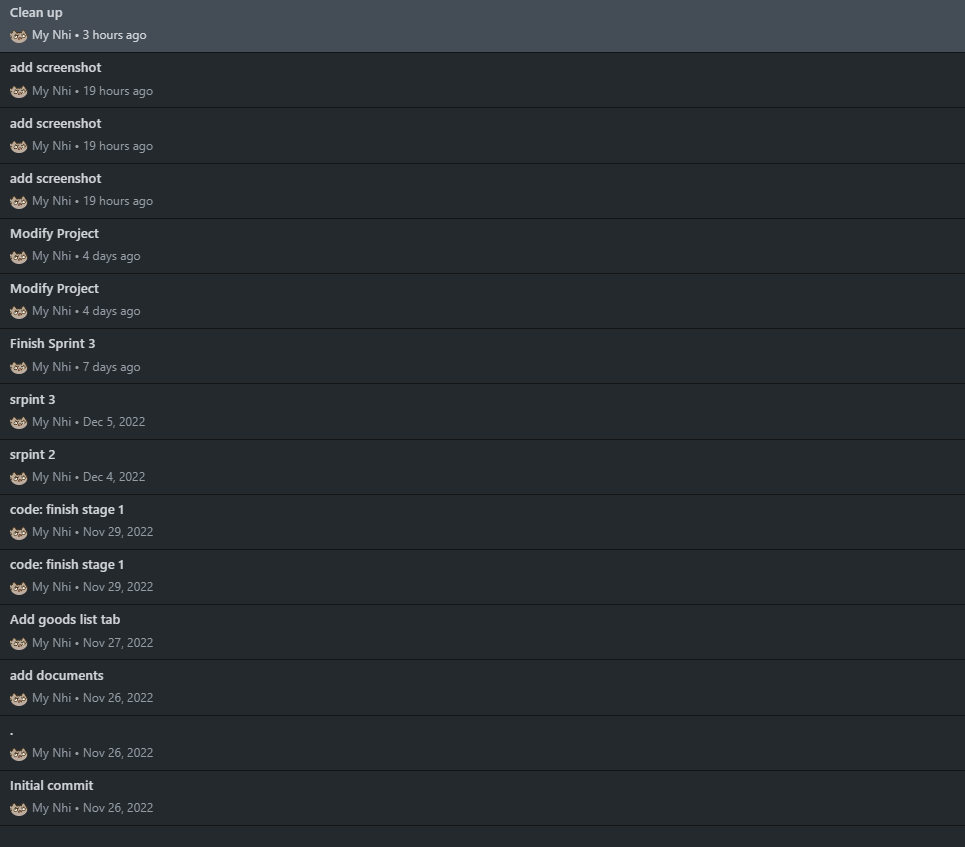
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Testcase description** | **Testcase procedure** | **Expected Result** | **Test**  **Result** |
| **1** | Test quantity textBox can only accept number | 1. Quantity textBox is empty 2. Input some text | Message displays “ ” | Not Tested |
| **2** | Test comboBox product name is  empty | 1. Product name comboBox is empty 2. Click “Add” | Message displays “” | Not Tested |
| **3** | Quantity textBox is empty | 1. Quantity textBox is empty 2. Click “Add” | Message displays  “” | Not Tested |
| **4** | Product list on create receive note is empty | 1. Product list is empty 2. Click “Create Oder” | Message displays  “” | Not Tested |
| **5** | Click on order from Order list | 1. Click on order from Order list | Order’s product detail listView display the correct products of the  clicked Order | Not Tested |
| **6** | Month comboBox is empty | 1. Month comboBox is not selected 2. Click “Statistical” | Message display  “” | Not Tested |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **7** | Month comboBox is empty | 1. Month comboBox is not selected 2. Choose   delivery/receive comboBox | Message display “” | Not Tested |

6.2 Unit Test (NOT TESTED)

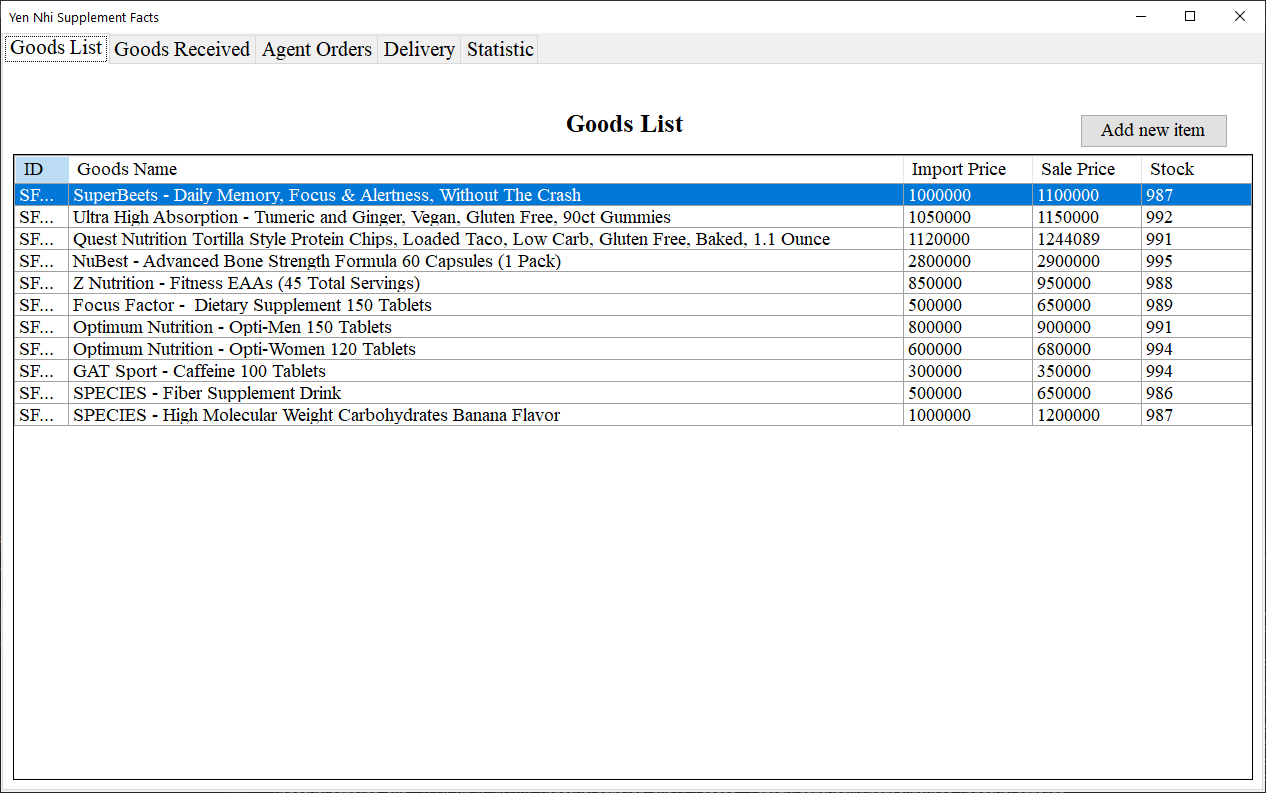
6.3 GIT

****

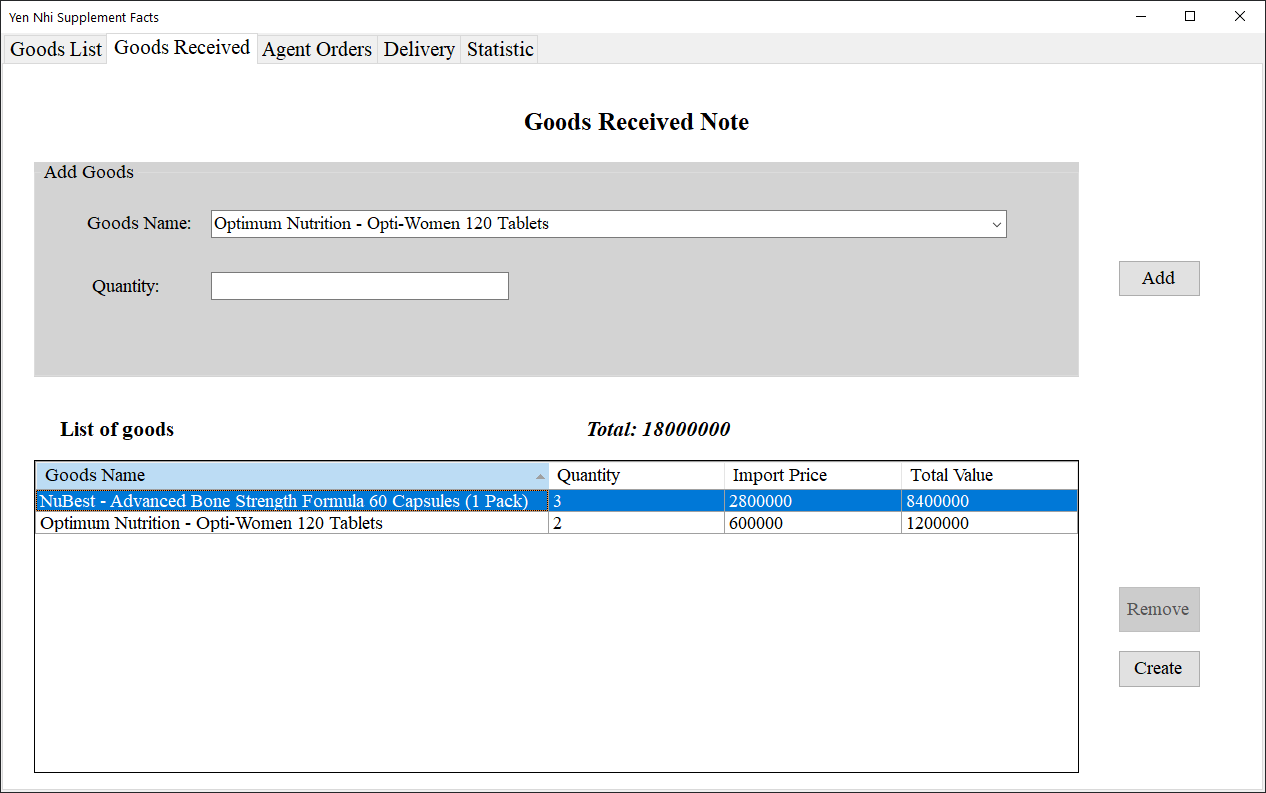
****

# CHAPTER VII – DEMO

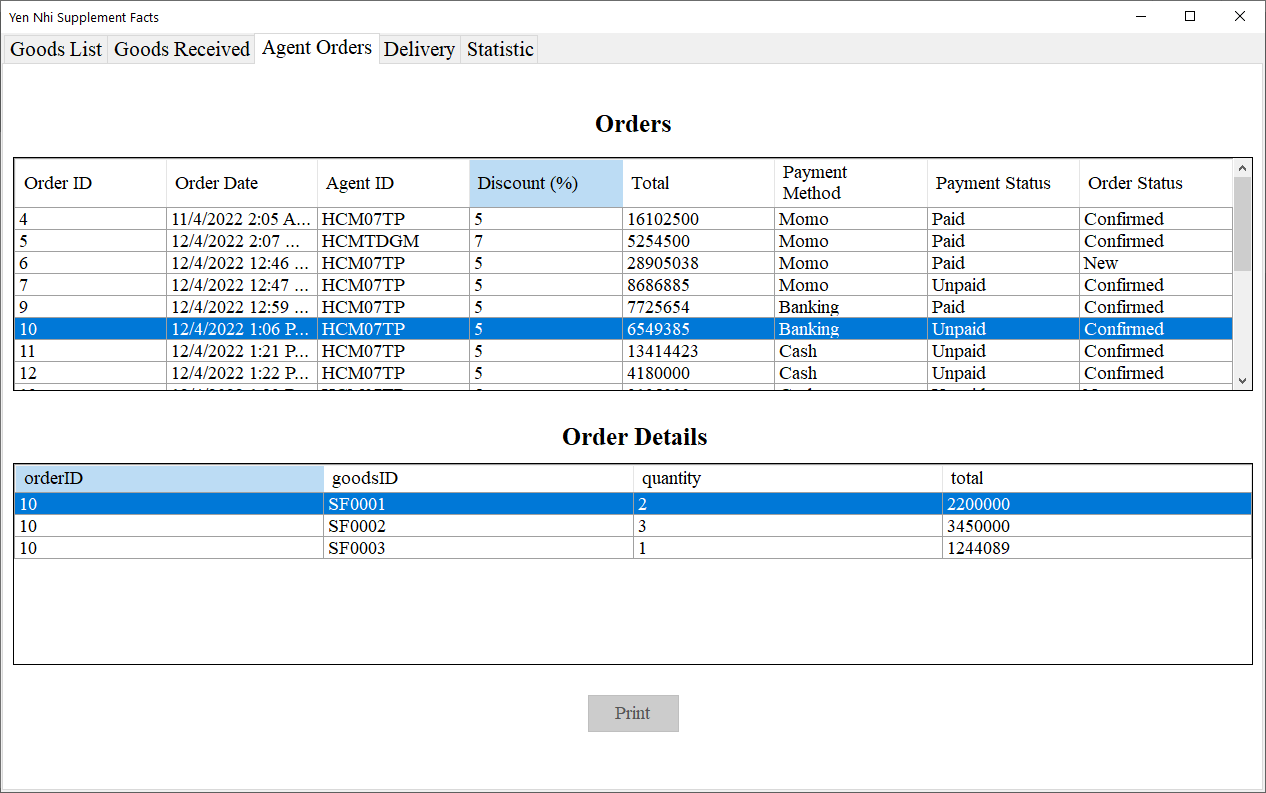
* WinForm Goods List



* WinForm Create Goods Received Note



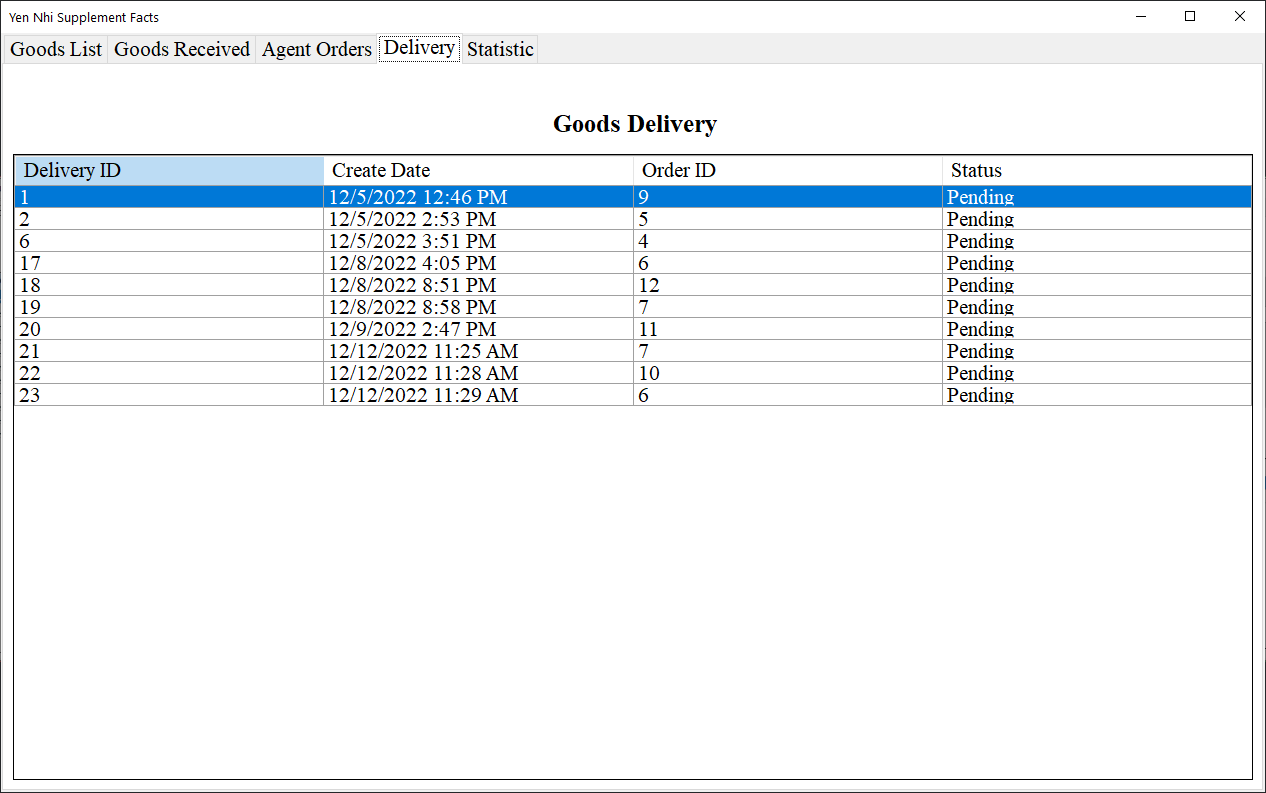
* WinForm Agent Oders List and create Goods Delivery Note



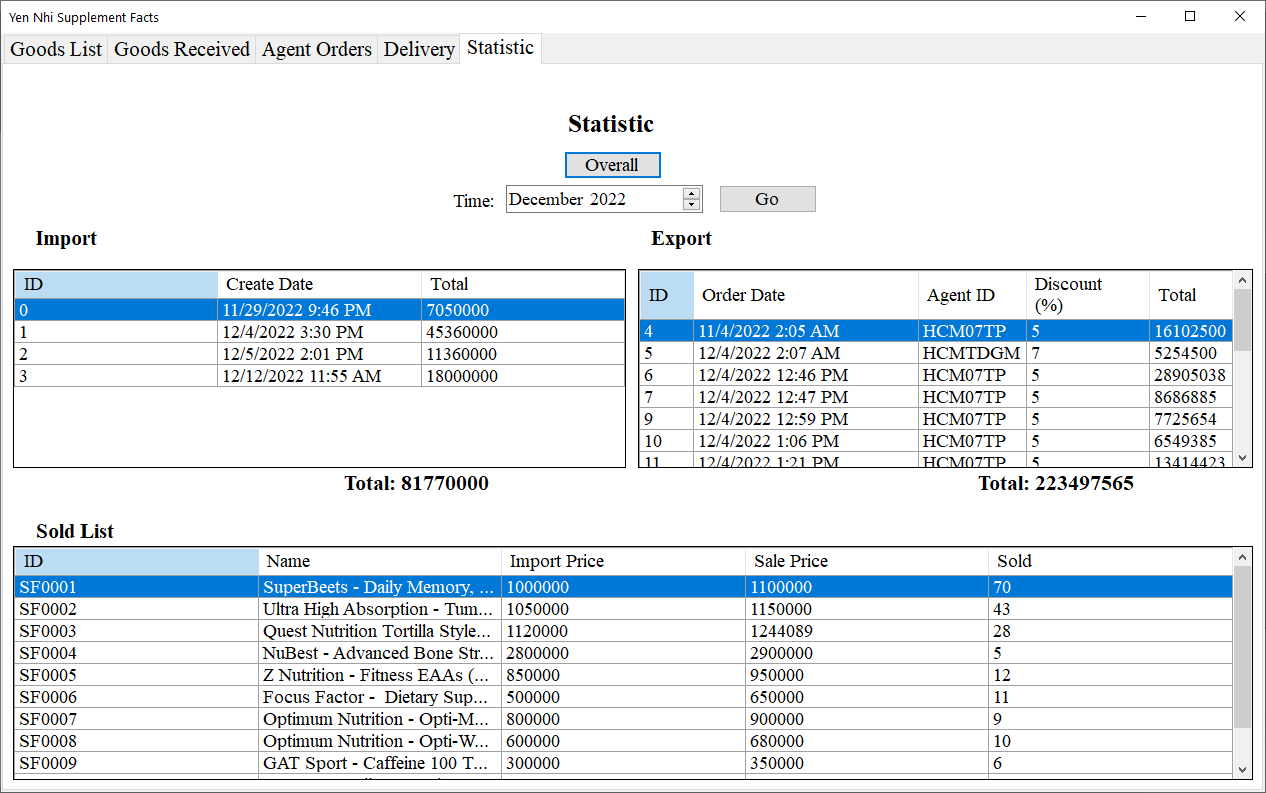
* Example for print goods delivery note



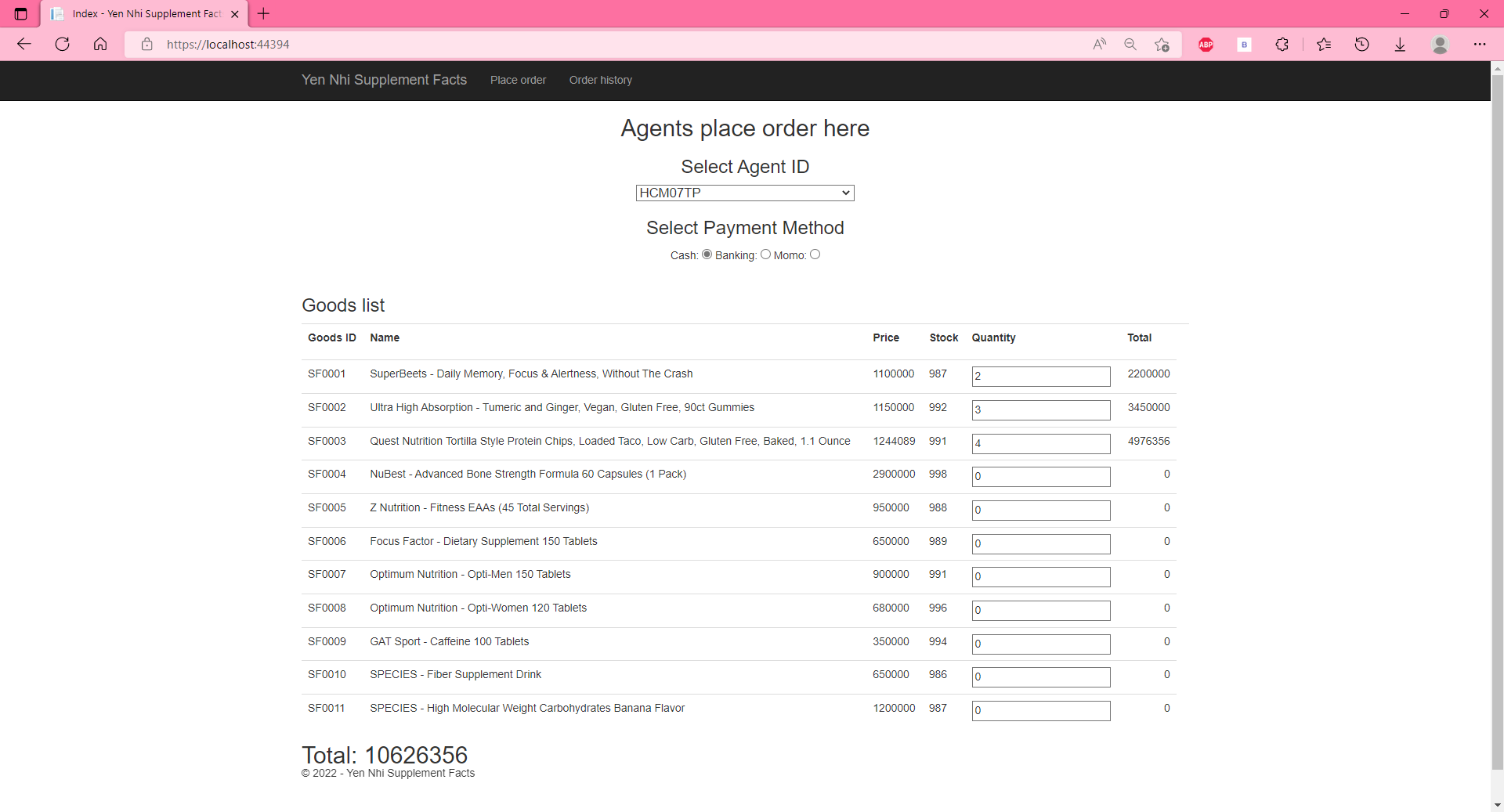
* WinForm Goods Delivery List



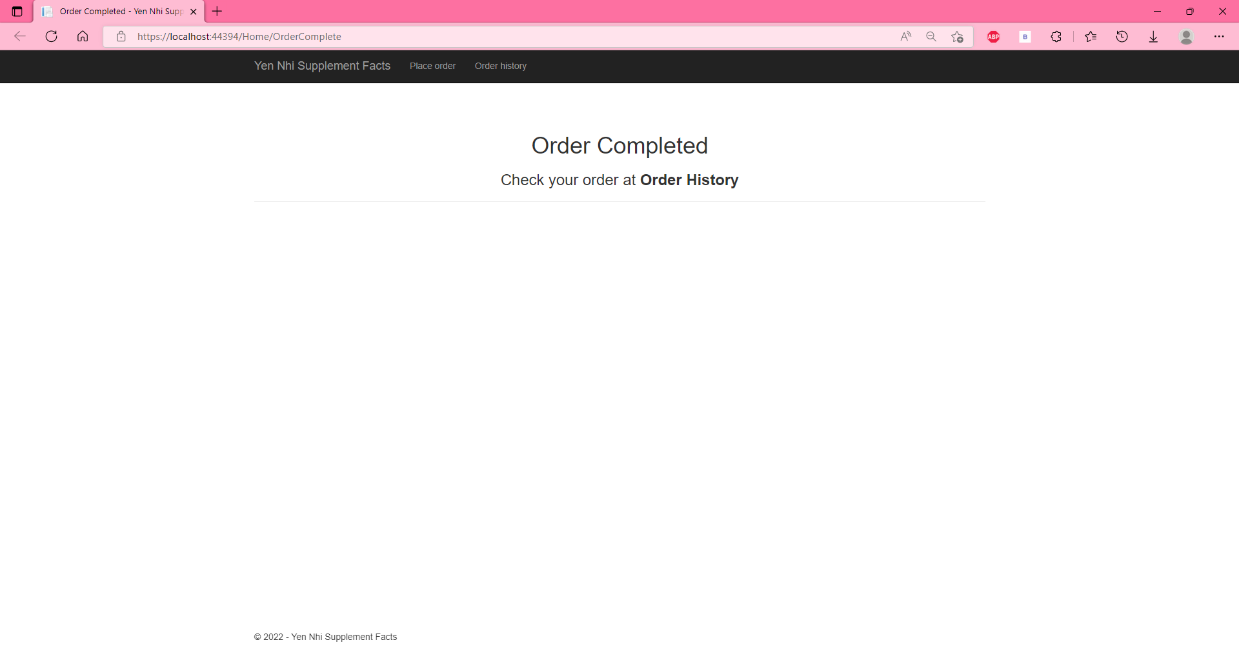
* Statisticby month



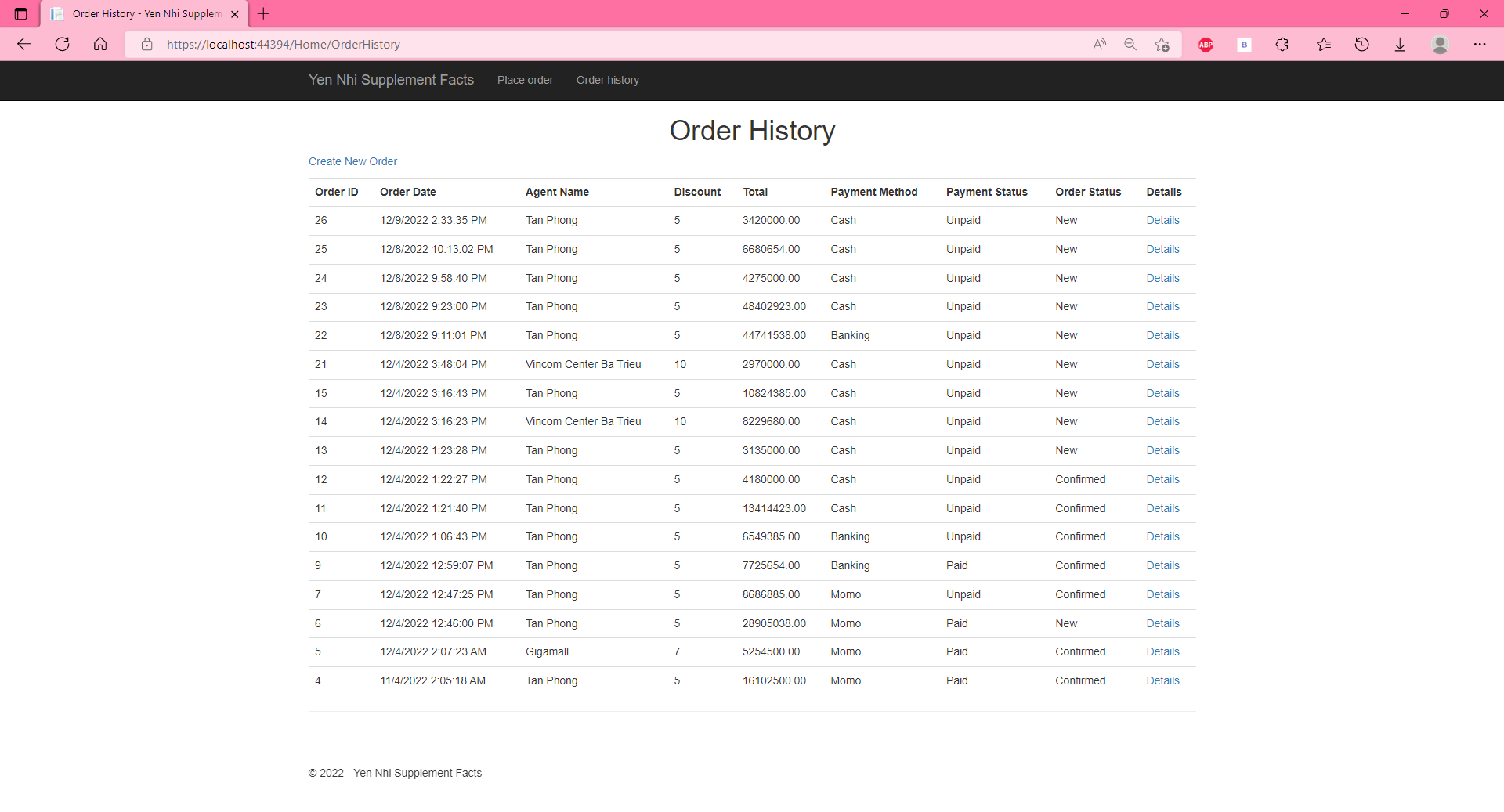
* .NET MVC Order Form



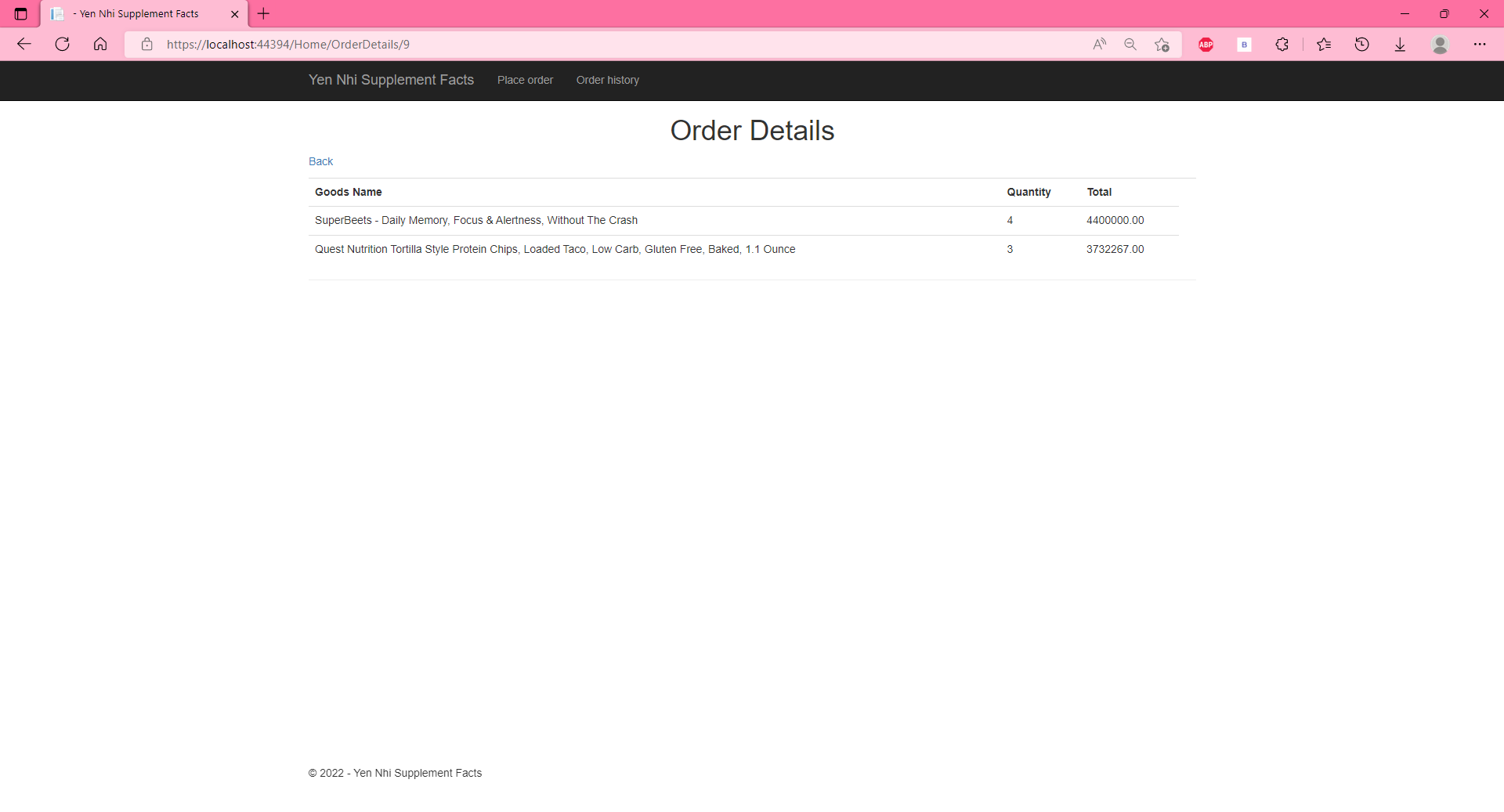
* .NET MVC Agent Order complete



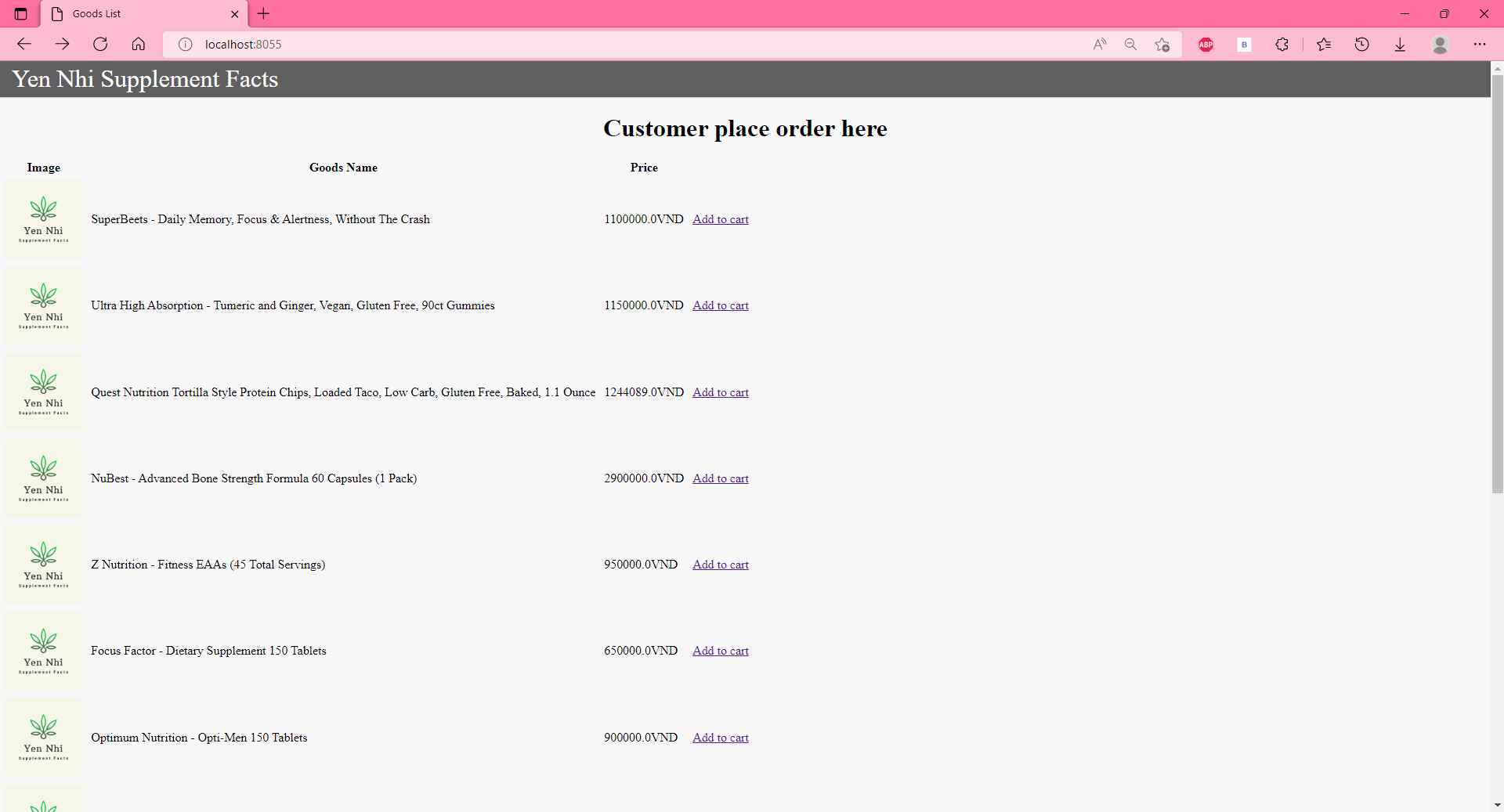
* .NET MVC Agent View Order History



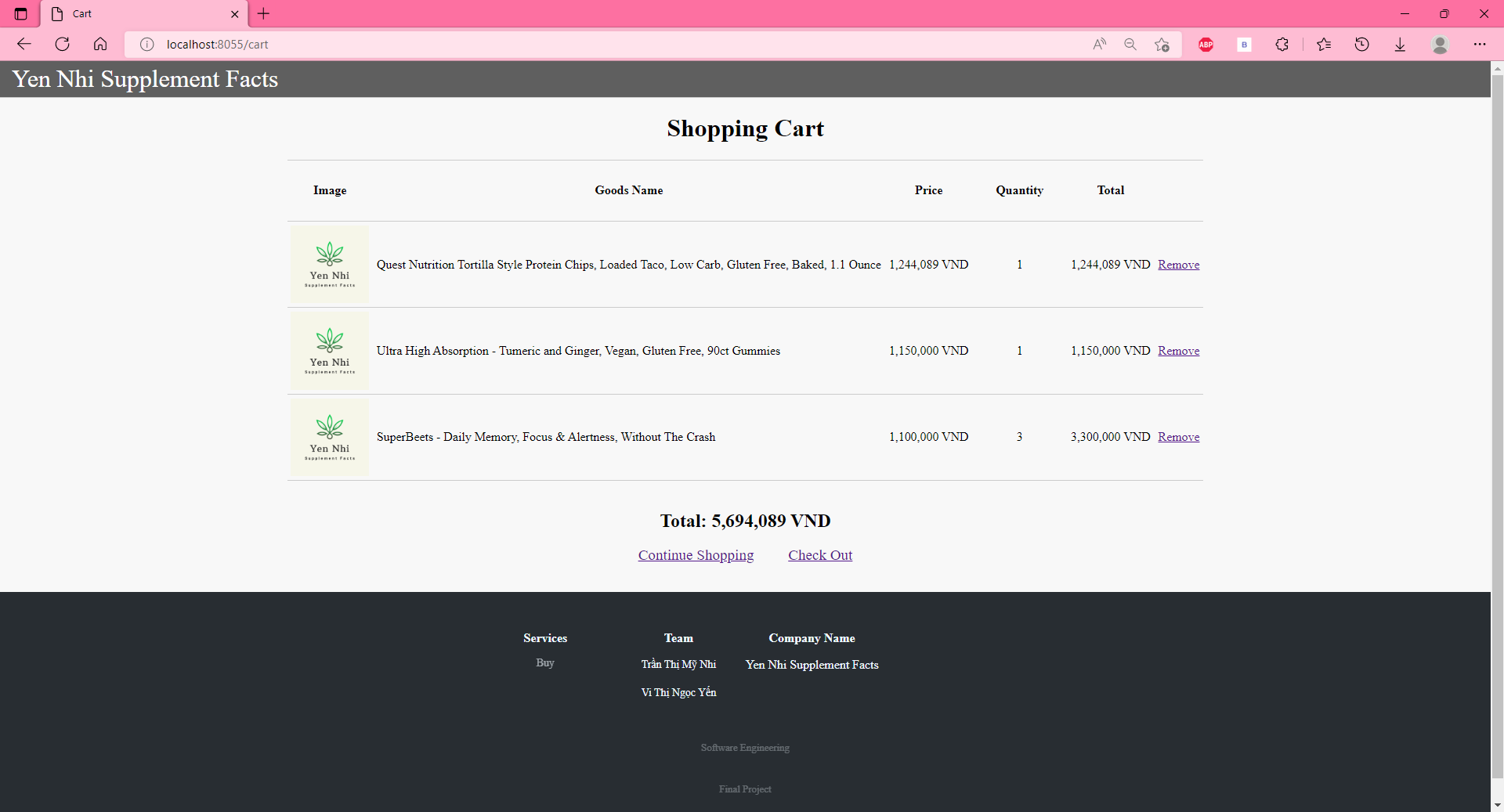
* .NET MVC Agent View Order Details



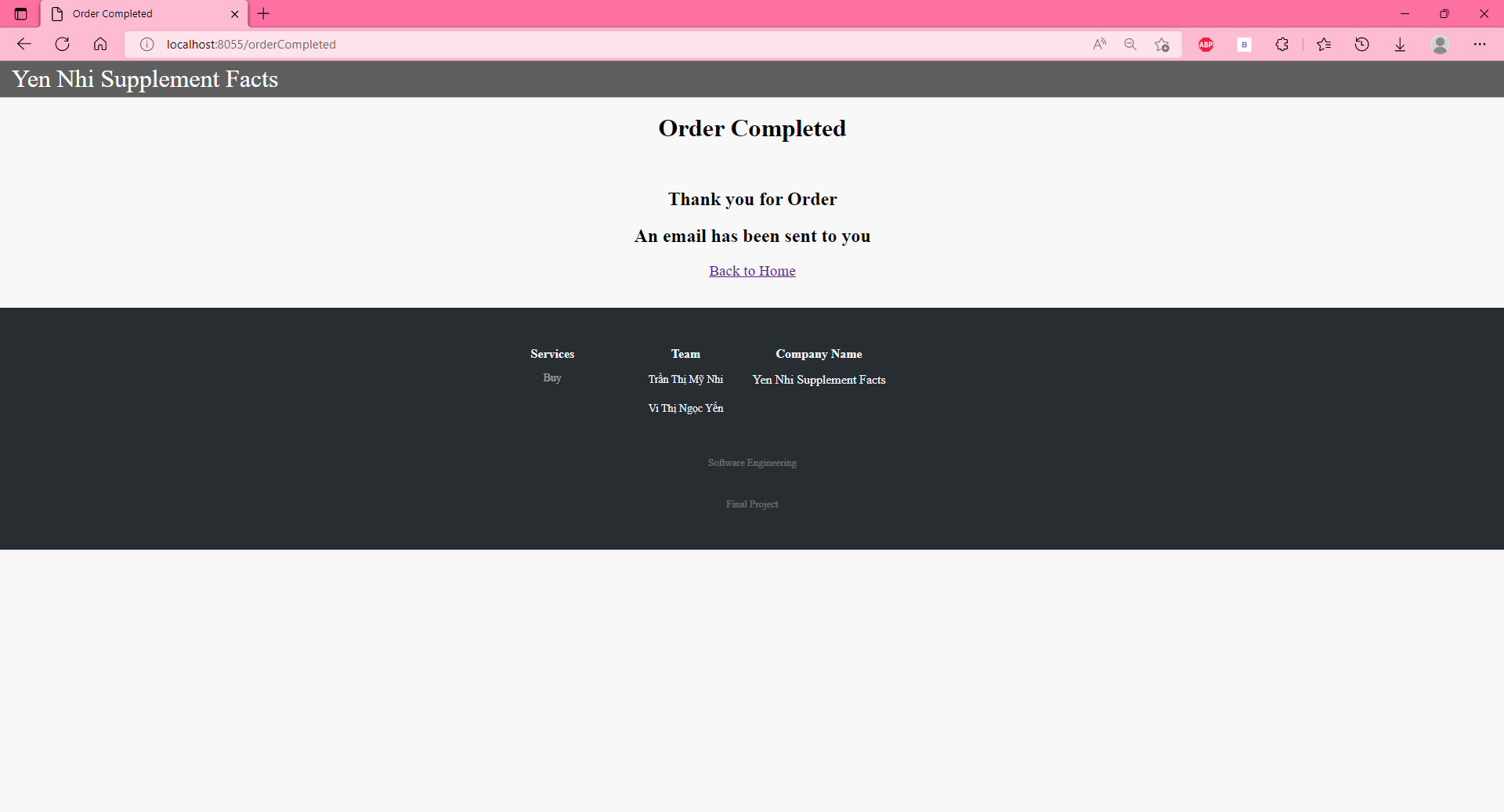
* Sping MVC Customer Order Form



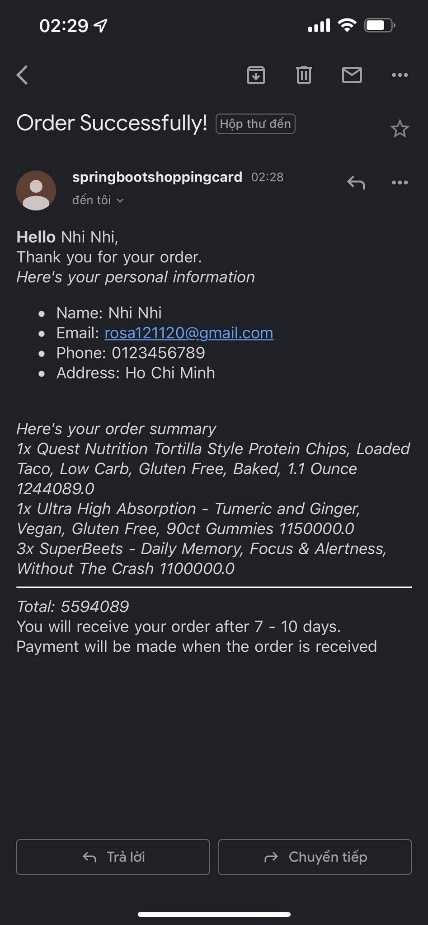
* Sping MVC Customer Shopping Cart



* Spring MVC Customer Order Complete



* Spring MVC Customer Received Mail After Completed Order

****

REFERENCES

1. Print to PDF [PDF-Examples/Getting Started/Windows Forms/Create-PDF-with-basic-elements/Create-PDF-with-basic-elements at master · SyncfusionExamples/PDF-Examples (github.com)](https://github.com/SyncfusionExamples/PDF-Examples/tree/master/Getting%20Started/Windows%20Forms/Create-PDF-with-basic-elements/Create-PDF-with-basic-elements)
2. [Spring Boot Connect to Microsoft SQL Server Examples (codejava.net)](https://www.codejava.net/frameworks/spring-boot/connect-to-microsoft-sql-server-examples)